RICE UNIVERSITY

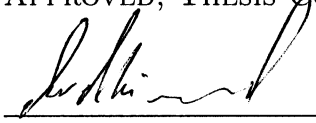
Computational models of signaling processes in cells with applications: Influence of stochastic and spatial effects

by

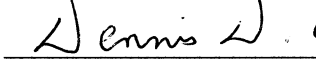
Roberto Bertolusso

A Thesis Submitted
in Partial Fulfillment of the
Requirements for the Degree
Doctor of Philosophy

Approved, Thesis Committee:

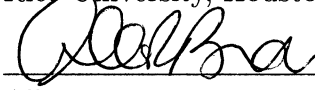


Marek Kimmel, Chair
Professor of Statistics and Bioengineering
Rice University, Houston, TX

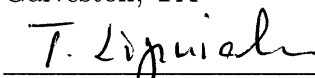


Dennis E. Cox
Professor of Statistics
Rice University, Houston, TX

Michael W. Deem
Professor of Bioengineering
Rice University, Houston, TX



Allan R. Brasier
Professor of Endocrinology
University of Texas Medical Branch,
Galveston, TX



Tomasz Lipniacki
Associate Professor
Institute of Fundamental Technological
Research, PAN, Warsaw, Poland

Houston, Texas
August, 2011

ABSTRACT

Computational models of signaling processes in cells with applications: Influence of stochastic and spatial effects

by

Roberto Bertolusso

The usual approach to the study of signaling pathways in biological systems is to assume that high numbers of cells and of perfectly mixed molecules within cells are involved. To study the temporal evolution of the system averaged over the cell population, ordinary differential equations are usually used. However, this approach has been shown to be inadequate if few copies of molecules and/or cells are present.

In such situation, a stochastic or a hybrid stochastic/deterministic approach needs to be used. Moreover, considering a perfectly mixed system in cases where spatial effects are present can be an over-simplifying assumption. This can be corrected by adding diffusion terms to the ordinary differential equations describing chemical reactions and proliferation kinetics. However, there exist cases in which both stochastic and spatial effects have to be considered.

We study the relevance of differential equations, stochastic Gillespie algorithm, and deterministic and stochastic reaction-diffusion models for the study of important biological processes, such as viral infection and early carcinogenesis. To that end we have developed two optimized libraries of C functions for R (r-project.org) to simulate biological systems using Petri Nets, in a pure deterministic, pure stochastic, or

hybrid deterministic/stochastic fashion, with and without spatial effects. We discuss our findings in the terms of specific biological systems including signaling in innate immune response, early carcinogenesis and spatial spread of viral infection.

A mi amor Adriana, símbolo de las cosas más lindas de la vida;

a mi mamma Luisanna, que me apoyó siempre y me brindó todo su amor;

a mi padre Renato, un gran hombre que me dio vida y apenas pude conocer;

a mi padre Horacio, que me crió y fue un ejemplo de honestidad y rectitud;

y a la Tata, de la que aprendí que no hay una sola realidad.

Contents

Illustrations

Tables

Chapter 1

Introduction

1.1 Summary

The aim of this research is to study stochastic and spatial effects in signaling pathways in biological systems and their interaction and particularly to determine when these stochastic effects are important.

The usual approach is to assume that high numbers of perfectly mixed cells and molecules within cells are involved and that they are in thermal equilibrium, and to study effects averaged over the cell and molecule population. To that end, it is customary to approximate the temporal evolution of the system with ordinary differential equations.

However, this approach has been shown to be inadequate for mesoscopic systems, where few copy numbers of cells and/or molecules are present. In these cases a stochastic approach, or a hybrid stochastic/deterministic one, needs to be used. Moreover, considering a perfectly mixed system in cases where spatial effects are present can be, in certain cases, an over-simplifying assumption. This can be corrected, for example, by adding diffusion terms to the ordinary differential equations describing chemical reactions and proliferation kinetics. However, most cases are still exclusively treated in a deterministic manner.

We are interested in studying the relevance of stochastic reaction-diffusion processes that may be applicable to the study of important biological systems, such as viral infection spread or early carcinogenesis.

As any but the simplest mass-action systems are mathematically intractable, we have developed optimized software, based on Petri nets methodology, to numerically solve reaction-only and reaction-diffusion systems, employing deterministic, stochas-

tic, and hybrid methods.

1.2 Thesis organization

The current thesis is organized in the following way:

- Part I concerns systems where spatial effects can be neglected, and it is covered in Chapters 2 to 5 (starting on page 6).
- Part II concerns systems where spatial effects are relevant, and it is covered in Chapters 6 to 9 (starting on page 75).
- We discuss our findings in Chapter 10 (starting on page 155).
- Additional material is presented in the Appendix (starting on page 159).

1.3 Chapter organization

Chapters are organized in the following way:

- In Chapter 2 (page 6) we include background information concerning deterministic and stochastic simulations without a spatial component. It consists of four parts. In part one we introduce modeling of signaling pathways in cells and the paradigm of Petri nets as a systematic way to model biological systems. In part two we treat the molecular approach to chemical kinetics, presenting both the deterministic and stochastic methods of solving a set of coupled chemical reactions. In part three we derive the master equation and study when the deterministic solution of a set of ordinary differential equations coincides with the expected value of the stochastic kinetic model. In part four we review the current approaches to algorithmical solutions of deterministic, stochastic, and hybrid stochastic/deterministic systems.

- In Chapter 3 (page 26) we introduce biological background. We present viruses, the process of infection, the innate immune system response, and major signaling pathways associated. Key proteins, cytokines and receptors are summarized.

- In Chapter 4 (page 41) we introduce the software library bioPN (biological Petri Nets) and its applications. It consists of three parts. In part one we describe bioPN. In part two we validate numerical methods used in bioPN against analytical solutions. In part three we compare deterministic and stochastic solutions for different simple pathways, and we show a case where bi-stability arises. This latter was published paper about crosstalk between the transcription factors NF-κB and p53 [58].

- In Chapter 5 (page 61) we presents current work on a model of cellular sensing of double stranded RNA, which is itself a model of viral infection, by RIG-I protein and TLR3 receptor. It consists of six parts. In part one we introduce the pathway. In part two we describe experiments. In part three we describe experimental results. In part four we introduce the current version of the associated computational model. In part five we describe simulations, protocol used, and simulation results. In part six we compare the outcomes of the deterministic run and of the average of runs of stochastic simulations.

- In Chapter 6 (page 75) we introduce reaction-diffusion systems. It consists of two parts. In part one we discuss the importance of considering spatial effects, such as diffusion, as well as incorporating stochasticity in models where a deterministic approach is not a viable solution. In part two we summarize the mathematical background relevant to modeling of reaction-diffusion systems, and deterministic and stochastic numerical approximations.

- In Chapter 7 (page 85) we present sbioPN (spatial bioPN), a software library developed to solve reaction-diffusion systems in deterministic, stochastic, and

hybrid deterministic/stochastic manner. It consists of six parts. In part one we discuss the limitations of bioPN to simulate reaction-diffusion systems. In part two we present sbioPN and its characteristics. In part three we explain the structures needed to define a reaction-diffusion model in sbioPN. In part four we compare diffusion in deterministic and stochastic settings. In part five we make some considerations about compartment size in cubic grids. In part six we shows two applications from literature simulated with sbioPN.

- In Chapter 8 (page 101) we present the stochastic version of an original one-dimensional reaction-diffusion model of early carcinogenesis that exhibits Turing instabilities. It consists of five parts. In part one we introduce the model. In part two we describe the deterministic and stochastic approaches employed. In part three we present results of both approaches and compare them. In part four we study the behavior of the system for different rates of diffusion. In part five we treat additional material of technical nature.

- In Chapter 9 (page 127) we compare deterministic and stochastic reaction-diffusion approaches in a two-dimensional model of viral infection with interferon production. It consists of five parts. In part one we explain the motivation for the study. In part two we introduce experimental conditions and how boundary conditions are treated. In part three we introduce the full model. In part four we construct the full model incrementally and we compare results of deterministic and stochastic systems at each step. In part five we detail the set of reactions of the full model.

Part I

Mathematical and biological background, comparison of deterministic and stochastic effects using bioPN modeling, and the IRF3/NF-κB model

Chapter 2

Mathematical modeling background

In this chapter we include background information concerning deterministic and stochastic simulations without a spatial component. It consists of four parts. In part one we introduce modeling of signaling pathways in cells and the paradigm of Petri nets as a systematic way to model biological systems. In part two we treat the molecular approach to chemical kinetics, presenting both the deterministic and stochastic methods of solving a set of coupled chemical reactions. In part three we derive the master equation and study when the deterministic solution of a set of ordinary differential equations coincides with the expected value of the stochastic kinetic model. In part four we review the current approaches to algorithmical solutions of deterministic, stochastic, and hybrid stochastic/deterministic systems.

2.1 Modeling of signaling pathways in cells

Models are created as a way to describe, in a system of interest, its elements and their interactions. Applied to molecular cell biology, a model can be used to describe mechanisms such as gene regulation, transcription, translation, positive and negative feedback, diffusion of components, and so forth.

We start by presenting a simple model of gene auto-regulation suitable for procaryotic cells [69]:

$$\begin{aligned} \mathrm{Gi} &\Longleftrightarrow \mathrm{Ga} + \mathrm{Dimer} & \qquad \mathrm{mRNA} &\longrightarrow \otimes \\ \mathrm{Ga} &\longrightarrow \mathrm{Ga} + \mathrm{mRNA} & \qquad \mathrm{Protein} &\longrightarrow \otimes \\ \mathrm{mRNA} &\longrightarrow \mathrm{mRNA} + \mathrm{Protein} & \qquad 2\ \mathrm{Protein} &\Longleftrightarrow \mathrm{Dimer} \end{aligned}$$

where symbol $\otimes$ denotes that the product is removed from the system. The same

symbol on the left of the reaction denotes that the product is added to the system.

There is a total of 6 chemical reactions, two of them reversible (reactions one and six) and the rest irreversible:

- The first reaction (reversible) represents activation of inactive gene Gi once it dissociates from an inhibitor, which is the dimer of the protein expressed by the active gene Ga. The reverse binding reaction inactivates the gene (which constitutes a negative feedback).
- The second reaction represents transcription of the gene that produces mRNA molecules without being depleted (Ga is on both sides of the expression).
- The third reaction is translation of the mRNA, producing the corresponding protein.
- The fourth and fifth reactions represent the degradation of the mRNA transcripts and the proteins.
- Finally, the sixth reaction (reversible) represents dimerization of the protein, where two molecules of protein join, and the reverse reaction where the dimer dissociates into two proteins.

2.2 Petri Nets

Petri nets have been successfully applied to a wide range of applications, mainly related to modeling of man-made systems. More recently they have been adopted to study molecular networks. [9]. A complete introduction can be found in Ch. 2 of Wilkinson [69].

In short, a Petri Net is mathematical framework for systems modeling together with an associated graphical representation [49, 60]. It is easy to represent the above

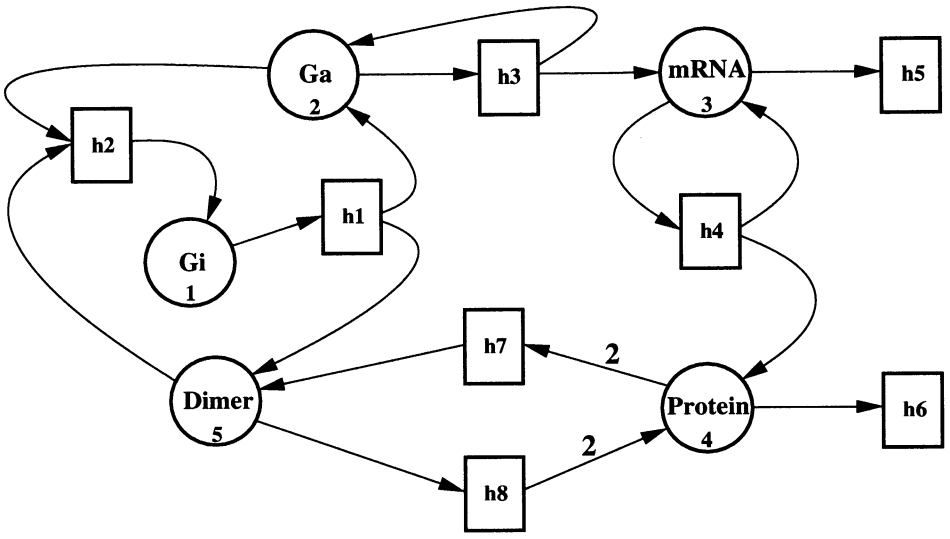


Figure 2.1 : Petri net graphical representation of the biological network of Section 2.2.

model as a Petri Net. The only modification needed is decomposing reversible reactions into pairs of irreversible reactions:

$$\text{Gi} \xrightarrow{c_1} \text{Ga} + \text{Dimer} \tag{2.1}$$

$$\text{Ga} + \text{Dimer} \xrightarrow{c_2} \text{Gi} \tag{2.2}$$

$$\text{Ga} \xrightarrow{c_3} \text{Ga} + \text{mRNA} \tag{2.3}$$

$$\text{mRNA} \xrightarrow{c_4} \text{mRNA} + \text{Protein} \tag{2.4}$$

$$\text{mRNA} \xrightarrow{c_5} \otimes \tag{2.5}$$

$$\text{Protein} \xrightarrow{c_6} \otimes \tag{2.6}$$

$$2\ \text{Protein} \xrightarrow{c_7} \text{Dimer} \tag{2.7}$$

$$\text{Dimer} \xrightarrow{c_8} 2\ \text{Protein} \tag{2.8}$$

The corresponding Petri Net graphical representation is depicted in Figure 2.1. Circles represent chemical species (places in Petri net terminology), and boxes reactions (transitions in Petri net terminology).

$$\text{Gi} \xrightarrow{c_1} \text{Ga} + \text{Dimer}$$

$$\text{Pre}[1,] = (1 \quad 0 \quad 0 \quad 0 \quad 0)$$
$$\text{Post}[1,] = (0 \quad 1 \quad 0 \quad 0 \quad 1)$$

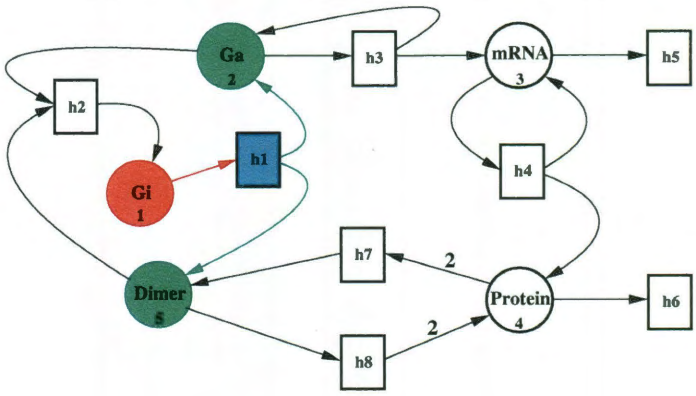


Figure 2.2 : Reaction 1: Gene activation. The reaction box is denoted in blue. Incoming arc (in red) denotes the stoichiometry of the left side of the chemical reaction (1 is assumed if empty) and it is entered in the pre matrix on the column corresponding to the reactant, Gi in this case, from which the arc starts. The arcs leaving the reaction box (in green) denote the stoichiometry of the right side of the chemical reaction (also 1 in this case) and are entered in the post matrix. In both pre and post, the row in which the stoichiometries are entered correspond to the actual reaction.

The arcs leading into each box denote reactants, and the arcs leading out of each box denote products. The numbers accompanying the arcs denote reaction stoichiometries (by a convention, empty implies stoichiometry of 1). Each node has an integer number of tokens associated with it, and the collection of all token numbers at a certain time is known as the current state (marking in Petri net terminology) of the system.

The formal definition of a Petri net follows:

Definition 2.1 A Petri Net, PN, is an n-tuple $(Pl, Tr, Pre, Post, M)$, where $Pl = \{pl_1, \cdots, pl_u\}$, $(u > 0)$ is a finite set of places (nodes, species), $Tr = \{tr_1, \cdots, tr_v\}$, $(v > 0)$ is a finite set of transitions (reactions), and $Pl \cap Tr = \otimes$.

Pre is a $v \times u$ integer matrix containing the weights (stoichiometries) of the arcs from places to transitions (the element (i, j) of this matrix is the weight of the arc from place j to transition i), and $Post$ is a $v \times u$ integer matrix containing the weights (stoichiometries) of the arcs from transitions to places (the element (i, j) of this matrix is the weight of the arc from transition i to place j).

M is a u-dimensional integer vector representing the current marking of the net (i.e. the current state of the system).

Figure 2.2 shows the Petri net with the first reaction highlighted. Figures with the remaining seven reactions highlighted can be found in Appendix A, from page 159 to 161.

The complete Petri Net mathematical representation of the previous example has the form:

$$PN = (Pl, Tr, Pre, Post, M(t))$$

$$Pl = \begin{pmatrix} \text{Gi} \\ \text{Ga} \\ \text{mRNA} \\ \text{Protein} \\ \text{Dimer} \end{pmatrix} \quad Tr = \begin{pmatrix} \text{Gene activation} \\ \text{Gene inactivation} \\ \text{Transcription} \\ \text{Translation} \\ \text{mRNA degradation} \\ \text{Protein degradation} \\ \text{Protein dimerization} \\ \text{Dimer dissociation} \end{pmatrix} \quad M(t) = \begin{pmatrix} m_1(t) \\ m_2(t) \\ m_3(t) \\ m_4(t) \\ m_5(t) \end{pmatrix}$$

$$Pre = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} \quad Post = \begin{pmatrix} 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 2 & 0 \end{pmatrix}$$

2.3 Molecular approach to chemical kinetics

In molecular approach, the number of molecules (or their mass) is used as a measure of the amount of each chemical species. We will not use concentrations to measure chemical species, although it is a common practice when using deterministic methods, as they cannot be directly compared to results of stochastic methods.

However, the conversion between concentrations and number of molecules is straightforward, and we present here the relation between them. Suppose M is the number of molecules of a certain species from which we know the concentration $[M]$ in moles per liter. Then:

$$M = n_A[M]V \tag{2.9}$$

where $n_A \approx 6.023 \cdot 10^{23}$ is the Avogadro's number and V is the volume in liters.

When studying deterministic methods, the quantities are real numbers, and when studying stochastic methods, the quantities are integers.

2.3.1 Mass-action kinetics

The deterministic time evolution of the amount of chemical species in the system described by the set of coupled chemical reactions defined by Equations 2.1 to 2.8 is obtained by solving the following system of ordinary differential equations (ODE):

$$\dot{I} = -c_1 I + c_2 AD \tag{2.10}$$

$$\dot{A} = c_1 I - c_2 AD \tag{2.11}$$

$$\dot{M} = c_3 A - c_5 M \tag{2.12}$$

$$\dot{P} = c_4 M - c_6 P - 2c_7 P^2 + 2c_8 D \tag{2.13}$$

$$\dot{D} = c_1 I - c_2 AD + c_7 P^2 - c_8 D \tag{2.14}$$

where $I = \text{Gi}$, $A = \text{Ga}$, $M = \text{mRNA}$, $P = \text{Protein}$, and $D = \text{Dimer}$, and the commonly used notation $\dot{M} = dM/dt$ for the derivative with respect to time is adopted.

There are several numerical methods to solve the above system of ODEs, in increasing order of complexity and precision. We will illustrate here the simplest of them, called Euler method, by applying it to the first equation.

We replace the differential equation by its associated difference equation:

$$\dot{I} = dI/dt \approx \Delta I/\Delta t = (I_{n+1} - I_n)/(t_{n+1} - t_n) \tag{2.15}$$

and substitute in Equation 2.10:

$$\begin{aligned} (I_{n+1} - I_n)/(t_{n+1} - t_n) &= -c_1 I + c_2 AD \\ I_{n+1} &= I_n + (-c_1 I + c_2 AD)(t_{n+1} - t_n) \end{aligned} \tag{2.16}$$

The same procedure is applied to Equations 2.11 to 2.14. The resulting system can be solved by iteration, for $n = 0, 1, \cdots, N$, starting from initial conditions I_0, A_0, M_0, P_0, and D_0. Euler method assumes step $t_{n+1} - t_n$ is constant.

We will introduce in Chapter 4 a software tool called bioPN, which solves such systems of equations using the more advanced method of Runge Kutta Dormand Prince 45 [11], that allows for dynamic time step adjustments.

It is important to note that the reaction constants c_i (used when considering numbers of molecules) may or not be the same as the ones that are used in the case of concentrations (usually denoted by k_i). Conversion between them also involves expressions containing n_A and V, but the laws are different according to the order of the reactions. We will summarize conversion between both systems for zero to second order reactions. In what follows, M is measured in moles per liter.

- Zero-order reactions:

  $\otimes \xrightarrow{k_i} \cdots \qquad c_i = n_A V k_i \qquad [c_i] = molecules/tu$

- First-order reactions:

  $M_j \xrightarrow{k_i} \cdots \qquad c_i = k_i \qquad [c_i] = 1/tu$

- Second-order reactions:

  $M_j + M_k \xrightarrow{k_i} \cdots \qquad c_i = k_i/(n_A V) \qquad [c_i] = 1/(molecules\ tu)$

  $2M_j \xrightarrow{k_i} \cdots \qquad c_i = 2k_i/(n_A V) \qquad [c_i] = 1/(molecules\ tu)$

where tu stands for time unit.

2.3.1.1 Volume change in a closed system

In a closed system, changes in volume of the container will affect systems measured either in concentrations or in number of molecules:

- If concentrations are used, any change in volume will affect the concentrations themselves. For example, if the volume doubles, the concentrations halve, while the rate constants k_i remain unaffected.

- If number of molecules are used, the opposite holds true. The number of molecules remain unaffected. Rate constants c_i, instead, may need correction. Units of volume do not appear in c_i; they vanish in the conversion process from k_i to c_i, as it can be verified by conducting a dimensional analysis. However, as they were obtained for a specific volume, they are sensitive to its change. For example, if the volume doubles, a zero order c_i doubles, and a second order c_i halves. A first order c_i remains unaffected.

The need of rate constant correction under volume changes will be important for the second part of this work, were reaction-diffusion processes are treated. However, in that case, we will be interested in partitioning the closed system in compartments, not in changing the volume of the container of the whole closed system. In addition to the correction of the corresponding c_i, number of molecules inside each smaller compartment will also need to be corrected to satisfy the constraint that the total

number of molecules after partitioning has to match the total number of molecules in the original non-partitioned closed system.

2.3.2 Mass-action stochastic kinetics

The deterministic approach to chemical kinetics fails to capture the discrete and stochastic nature of chemical kinetics at low concentrations. To develop a molecular approach, let us consider two perfectly spherical molecules X and Y which react if they collide with sufficient energy while they are moving randomly. If the container has constant volume and it is well mixed and in thermal equilibrium, the collision hazard is constant [19].

Each reaction has a stochastic rate constant c_i, and an associated rate law (propensity or hazard) $h_i(M(t), c_i)$. The form of $h_i(M(t), c_i)$ (and the units of the rate constant c_i) depends on the order of reaction:

- Zero-order reactions:

  $\otimes \xrightarrow{c_i} \cdots \qquad h_i(M(t), c_i) = c_i$

- First-order reactions:

  $M_j \xrightarrow{c_i} \cdots \qquad h_i(M(t), c_i) = c_i m_j(t)$

- Second-order reactions:

  $M_j + M_k \xrightarrow{c_i} \cdots \qquad h_i(M(t), c_i) = c_i m_j(t) m_k(t)$

  $2M_j \xrightarrow{c_i} \cdots \qquad h_i(M(t), c_i) = c_i \frac{m_j(t)(m_j(t)-1)}{2}$

where $m_l(t)$ is the number of reactant molecules of species l at time t.

Conditional on state M at time t, the probability that reaction i occurs in the time interval $[t, t + \Delta t]$ is given by $h_i(M(t), c_i)\Delta t + o(\Delta t)$ [19]. In the absence of any other reactions taking place, the time to such reaction is an exponentially distributed random variable with parameter $h_i(M(t), c_i)$ [19].

2.3.3 Kolmogorov's forward equation, or "chemical" master equation

The transition kernel of a Markov chain is defined as the following conditional probability:

$$p(x,t,x',t') \equiv \mathrm{P}\left(X(t+t') = x' | X(t) = x\right) \tag{2.17}$$

where x is the state of the process at continuous time t, for a finite number of states $S = 1, 2, \cdots, r$. If p does not depend on time, the process is called *homogeneous* and can be denoted $p(x, x', t')$, and expressed as a $r \times r$ transition matrix, $P(t')$. $P(0) = I$, the $r \times r$ identity matrix.

$P(t')$ is a transition matrix. As such, $P(t'_1 + t'_2) = P(t'_1)P(t'_2) = P(t'_2)P(t'_1)$.

The *transition rate matrix* is defined as:

$$Q \equiv \frac{d}{dt'}P(t')\bigg|_{t'=0} = \lim_{\delta t \to 0} \frac{P(\delta t) - P(0)}{\delta t} \tag{2.18}$$

The elements of Q are the *hazards* of moving to different states. Then the infinitesimal transition matrix is

$$P(\Delta t) = I + Q\,\Delta t + o(\Delta t) \tag{2.19}$$

$o(\Delta t)$ represents a quantity that is smaller than Δt, and it satisfies

$$\frac{o(\Delta t)}{\Delta t} \to 0 \text{ when } \Delta t \to 0 \tag{2.20}$$

$P(\Delta t)$ is a stochastic matrix. As such, it has non-negative elements and each row sums to one. As the off-diagonal elements of I are zero, the off-elements of $P(\Delta t)$ and $Q\,\Delta t$ must be the same up to a small with respect to Δt. Then the off-diagonal elements of Q have to be non-negative. Since the diagonal elements of P are bounded above by one and the diagonal elements of I are ones, the diagonal elements of Q must be negative. As the rows of both P and I sum to one, the rows of Q must sum to zero.

A probability row vector π will be stationary only if

$$\pi P(\Delta t) = \pi \quad \Longrightarrow \quad \pi(I + Q\ \Delta t + o(\Delta t)) = \pi \quad \Longrightarrow \quad \pi\ Q = 0 \tag{2.21}$$

Solving this last equation gives a stationary distribution for the system.

To derive $P(t)$ for a finite time t, we proceed as follows:

$$\begin{aligned} \frac{d}{dt}P(t) &= \lim_{\Delta t \to 0} \frac{P(t+\Delta t) - P(t)}{\Delta t} = \lim_{\Delta t \to 0} \frac{P(\Delta t)P(t) - P(t)}{\Delta t} \\ &= \lim_{\Delta t \to 0} \frac{P(\Delta t) - I}{\Delta t} P(t) = Q\ P(t) \end{aligned} \tag{2.22}$$

Then, given initial condition $P(0) = I$,

$$P(t) = \exp\{Q\ t\} \tag{2.23}$$

is a solution to the matrix differential equation

$$\frac{d}{dt}P(t) = Q\ P(t) \tag{2.24}$$

Note that $\exp\{\cdot\}$ is the matrix exponential function.

The corresponding component form is

$$\frac{d}{dt}p(i,j,t) = \sum_{k=1}^{r} q_{ik}\ p(k,j,t), \qquad i,j = 1,2,\cdots,r. \tag{2.25}$$

System (2.25) is known as the *Kolmogorov backward equations.*

Expansion of $P(t + \Delta t) = P(t)P(\Delta t)$ yields

$$\frac{d}{dt}P(t) = P(t)\ Q \tag{2.26}$$

the corresponding component form of which is

$$\frac{d}{dt}p(i,j,t) = \sum_{k=1}^{r} q_{kj}\; p(i,k,t), \qquad i,j = 1,2,\cdots,r. \tag{2.27}$$

System (2.27) is known as the *Kolmogorov forward equations.* In the language of computational biochemistry, these are the *"chemical" master equations.*

Wilkinson [69] (page 157) presents the Kolmogorov's forward equations in the form suitable for the stochastic Petri nets (SPN):

$$\begin{aligned}\frac{d}{dt}p(x_0,t_0,x,t) = \sum_{i=1}^{v}[&h_i(x - S^{(i)}, c_i)\; p(x_0,t_0,x - S^{(i)},t) + \\ &- h_i(x,c_i)\; p(x_0,t_0,x,t)], \qquad \forall t_0 \in \mathbb{R}, \quad x_0, x \in \mathcal{M}\end{aligned} \tag{2.28}$$

where $\mathcal{M}$ is the countable state space of the process, and $S^{(i)}$ the i-th row of the matrix S = post - pre.

2.3.4 Use of ODEs to solve a stochastic kinetic model

We are interested in exploring the relationship between the continuous deterministic approach (solution of the set of ODEs) and the expected value of the stochastic kinetic model. These two formulations yield the same results only in special cases.

The expected value of the state X_t of the process at time t is, by definition,

$$\mathrm{E}\left[X_t\right] = \sum_{x \in \mathcal{M}} x\; p(x,t) \tag{2.29}$$

Differentiating with respect to time yields

$$\frac{d}{dt}\mathrm{E}\left[X_t\right] = \frac{d}{dt}\sum_{x \in \mathcal{M}} x\; p(x,t) = \sum_{x \in \mathcal{M}} x\; \frac{d}{dt}p(x,t) \tag{2.30}$$

Detailed derivation is found in Appendix A, on page 18. We present here the re-

sult. After substituting the Kolmogorov's forward equation and performing algebraic manipulation, we obtain

$$\frac{d}{dt}\mathrm{E}\left[X_t\right] = \sum_{i=1}^{v} S^{(i)}\ \mathrm{E}\left[h_i(X_t, c_i)\right] \tag{2.31}$$

This set of differential equations cannot be solved directly, unless *all* reactions in the stochastic system have zero or first order mass action rate laws, in which case

$$\mathrm{E}\left[h_i(X_t, c_i)\right] = h_i\left(\mathrm{E}\left[X_t\right], c_i\right) \tag{2.32}$$

by linearity of expectation.

In this special case, after putting $y(t) = \mathrm{E}\left[X_t\right]$, we get

$$\frac{d}{dt}y(t) = \dot{y}(t) = \sum_{i=1}^{v} S^{(i)}\ h_i\left(\mathrm{E}\left[y(t)\right], c_i\right) \tag{2.33}$$

which is the ODE system for the deterministic model expressed in mass (not concentration) and using stochastic rate constants (as seen in Section 2.3.1). We will numerically verify this fact when comparing deterministic and stochastic methods in Section 4.3.1 on page 50.

As a conclusion, when *all* the reactions are of the zero or first order mass-action kinetics, the deterministic solution coincides with the expected value of the stochastic kinetic model.

2.4 Exact stochastic algorithms

Rules specified above define a time-continuous Markov chain, which might be simulated using any of the algorithms listed below.

2.4.1 Linear time algorithms

Gillespie, in the seminal paper [19], proposed the Direct Method, or SSA, to simulate chemical reactions. The algorithm is as follows:

1. Initialize the system (set $M(0)$).
2. Based on the current state, calculate $h_i(M(t), c_i)$ for all reactions.
3. Calculate $h_0(M(t), C) = \sum_{i=1}^{v} h_i(M(t), c_i)$, the combined reaction hazard.
4. Simulate the time to the next event, t', as a random variable with $Exp(h_0(M(t), C))$ distribution.
5. Put $t = t + t'$.
6. Assigning probability $h_i(M(t), c_i)/h_0(M(t), C)$ to reaction i $\{i = 1, \cdots, v\}$, simulate the index j of the reaction that occurred.
7. Update the state of the system $M(t)$ according to reaction j.
8. If $t < T_{max}$ return to step 2.

$M(t)$ is the state (marking) of the system at time t and $h_i(M(t), c_i)$ is the propensity of reaction i, that is a function of the current state and the reaction constant c_i. In its original formulation, the scaling of the SSA algorithm is linear in time with respect to the number of reactions both in the "update" steps 2 and 3, and in the "generation" step 6.

Gillespie also proposed the first reaction method, that replaces steps 3, 4 with

3. Simulate time to each event, t_i, as a $Exp(h_i(M(t), C))$ r.v.
4. Find $t' = t_j = \min\{h_i\}$.

and eliminates step 6.

Instead of using the total propensity to find the time to the next reaction, and then simulate which reaction happened, the first reaction method simulates when each reaction happens, and then chooses the one that occurs first. For this it has to use N random numbers per iteration, instead of the two needed by the SSA. The scaling of this method is also *O(N)*.

2.4.2 Logarithmic time algorithms

Gibson and Bruck proposed the next reaction method (NRM) [18]. This method is based on the first reaction method [19] and is considered by many the best approach for an exact method. Gibson and Bruck noted that using the memory-less property of the exponential distribution, the firing times of the reactions that did not occur can be rescaled.

In this way, the already generated corresponding random numbers can be reused. Consequently, after the initialization step only one random number needs to be generated per reaction that occurred, instead of two in the direct method. To reduce the number of computations, the method introduces a dependency graph in order to only update the propensities that need to be recalculated after the last reaction, as well as an indexed priority queue to determine which reaction will occur next. To implement the indexed priority queue two structures are needed; a binary tree structure with as many nodes as reactions, and an index to the memory position of each of the nodes. Both the update and the generation parts of the algorithm are performed in $log_2(N)$ steps.

In practice, this method is not always superior to an optimized version of the SSA method, also proposed by Gibson and Bruck [18]. The optimization of the SSA method consists in introducing the the same dependency graph used in the NRM, and introducing a binary tree that has at the leafs the propensities of the reactions,

which are accumulated moving towards the root. This ensures that both the update and the generation parts of the algorithm are performed in $log_2(N)$ steps.

The complete algorithm is the following:

1. Initialize
   (a) set initial conditions ($M(0)$).
   (b) Generate a dependency graph $\mathcal{G}$.
   (c) Calculate $h_i(M(t), c_i)$ for all reactions i.
   (d) For each i, simulate time for next event, τ_i, as a $Exp(h_i(M(t), C))$ r.v.
   (e) Store all the τ_i in an indexed priority queue $\mathcal{P}$.
2. Let j be the reaction where $\tau_j = min\{\tau_i\}$ (from $\mathcal{P}$).
3. Set $t = \tau_j$.
4. Update $M(t)$ according to reaction j.
5. From the dependency graph $\mathcal{G}$, find the subset of reactions whose propensities are modified by reaction j, and
   (a) For i in this subset, calculate $h_i(M(t), c_i)$.
   (b) For reaction j (the one that just fired), simulate time for next event, τ_j, as the sum of a $Exp(h_j(M(t), C))$ r.v. and t, and place it in the corresponding place of $\mathcal{P}$.
   (c) for $i \neq j$ in the subset, rescale the reaction times, $\tau_i = h_{i,old}(M(t), C)/h_{i,new}(M(t), C)(\tau_i - t) + t$, and put each of them in the corresponding place of $\mathcal{P}$.
6. if $t < T_{max}$ return to step 2.

2.4.3 Constant time algorithm

The Constant-Time Composition-Rejection algorithm [64] (SSA-CR) is suitable for the simulation of very large number of reactions. This algorithm has *O(1)* scaling in both generation and update times, independent of the number of reactions [64].

The generation portion of the SSA-CR algorithm has two stages, composition and rejection. In the composition stage a group of reactions is chosen according to the accumulated propensities of the reactions that belong to each group. This requires a random number and a selection algorithm similar to the one proposed by Gibson and Bruck [18] to optimize the reaction selection for the Gillespie algorithm. As the number of total groups can be made constant regardless of the number of reactions, this step takes constant time.

Once the group has been selected, the second stage uses the rejection algorithm. First a discrete uniform random number is generated to select one reaction inside the group. A second random number between zero and the maximum allowed propensity for reactions belonging to that group is generated and compared to the propensity of the selected reaction. If the generated number is less than the propensity, the rejection stage ends selecting this reaction. If not, the rejection part of the algorithm is iteratively repeated until the previous condition is satisfied.

If the groups are chosen in such a way that the upper and lower bounds of the propensities are such that $p_{upper}/p_{lower} = 2$, the expected number of iterations for the rejection stage is less than two [64]. This happens independently of the number of reactions in each group, substantiating the claim that the composition part is *O(1)*.

The update portion consists of comparing the old and new values of each reaction whose propensity is affected by the firing of the last reaction. If in each case the new propensity is still within the boundaries of its original group, only the group accumulated propensity and the global propensity need to be updated. If not, the reaction is excluded from the original group, and assigned to the corresponding group,

and both group propensities and global propensity are updated. If the old propensity is zero it means that the reaction is not in any group and it is inserted in one. If the new propensity is zero it means that the reaction has to be removed from the group.

In both cases the corresponding group accumulated propensity and global propensity are updated. In all cases the updating algorithm is an adaptation of the corresponding optimization of the Gillespie algorithms of Gibson and Bruck [18]. As long as the average number of reactions affected by the firing reaction does not grow continuously larger as the number of reactions grow, the update portion of the SSA-CR scales as *O(1)* [64].

2.5 Approximate stochastic algorithms

An important limitation of exact stochastic algorithms is the great amount of computer time that is needed to simulate a desired amount of system time [21], which becomes more evident as the number of different species and reactions increase, as well as the number of molecules per species. Approximate stochastic algorithms aim at making minor sacrifices in simulation accuracy in order to obtain major gains in simulation speed [21].

The fundamental difference between exact methods and approximate ones is that in the former the question is when will next reaction fire and in the latter how many reactions of each kind have fired on a specified amount of time.

Gillespie [20,21] proposed two approximate methods, the τ-leaping algorithm and the Chemical Langevin equation algorithm, reviewed below.

2.5.1 τ-leaping algorithm

This method assumes that it is possible to find a time interval τ where the reaction propensities $h_i(M(t), c_i)$ do not change "sensibly". If this is the case, the number of reactions of a certain kind that have fired in τ time-step is a Poisson distributed

random variable with parameter $h_i(M(t), c_i)\,\tau$.

The corresponding algorithm [21] is:

1. Initialize system ($M(0)$).
2. Determine a fixed time step τ such that the propensities will not change sensitively in that time step.
3. Based on current state, calculate $h_i(M(t), c_i)$ for all reactions i.
4. For each i, generate a $\mathcal{P}(h_i(M(t), c_i)\,\tau)$ r.v. p_i, and update the current state using $M_i(t+\tau) = M_i(t) + p_i$.
5. Put $t = t + \tau$.
6. if $t < T_{max}$ return to step 2.

The critical aspect of applying this and the following algorithms is choosing the right leaping step, for which several approaches have been proposed [7, 8, 22].

2.5.2 Chemical Langevin Equation (CLE) algorithm

Under the same assumptions of τ-leaping algorithm, Gillespie noted that, if the number of reactions of each kind in the specified amount of time is significantly high, the Poisson random variable can be further approximated using a normally distributed random variable.

This negligible loss of accuracy allows for an sensible increase in speed, as it is far less computationally demanding to generate a pseudo-random normal variate compared to a Poisson one. On the other hand, using this method introduces the use of continuous variables, instead of integer ones.

The corresponding algorithm [21] is:

1. Initialize system ($M(0)$).

2. Determine a fixed time step τ such that the propensities will not change sensitively in that time step.

3. Based on current state, calculate $h_i(M(t), c_i)$ for all reactions i.

4. For each i, generate a $\mathcal{N}(0,1)$ r.v. r_i, and update the current state using $M_i(t+\tau) = M_i(t) + h_i(M(t), c_i)\,\tau + \sqrt{h_i(M(t), c_i)\,\tau}\, r_i$.

5. Put $t = t + \tau$.

6. if $t < T_{max}$ return to step 2.

2.6 Deterministic methods

Often, biological systems are studied deterministically, usually using systems of ordinary differential equations. As already noted, the deterministic approach to chemical kinetics fails to capture the discrete and stochastic nature of chemical kinetics at low concentrations, and does not model the expected value of the stochastic process if reactions other than zero and first order are present. Generally, a linear integration with a predefined step, or Euler method, is used. More advanced integrators have been proposed to improve accuracy of integration. We have adopted the Runge Kutta modification proposed by Dormand and Prince [11], which allows adaptive time step selection, according to absolute and relative error bounds.

2.7 Hybrid stochastic/deterministic algorithms

Haseltine and Rawlings [27] noted that in most biological systems there are reactions that happen very often (*fast* reactions), while others rarely (*slow* reactions). They proposed the use of a hybrid algorithm that simulates slow reactions with the direct method, and the fast either deterministically or with CLE algorithm.

We will expand the treatment of the hybrid stochastic/deterministic algorithms in Chapter 4, where the bioPN library is presented.

Chapter 3

Biological Background

In this chapter we introduce biological background. We present viruses, the process of infection, the innate immune system response, and major signaling pathways associated. Key proteins, cytokines and receptors are summarized.

3.1 Biology of virus infection

The section on viral biology is mostly based on Chapter 24 of Alberts et al. [1].

3.1.1 Viruses

A virus is a very small and relatively simple particle that cannot be observed with a light microscope. Virus diameters are in the order of $10 - 450$nm ($1\text{nm} = 10^{-3}\mu\text{m} = 10^{-9}\text{m}$). As a rough comparison, on average prokaryotic cells (bacteria) are hundred times larger ($0.5 - 5\mu\text{m}$), while eukaryotic cells are thousand times larger ($5 - 20\mu\text{m}$).

Some viruses, as well as some bacterial, fungal and protozoan cells, are agents that cause infectious diseases. They are collectively known as pathogens, and usually exploit deleteriously the biological attributes of their hosts in order to survive and reproduce. Viral infections are responsible of acute and chronic diseases that range from the common cold to AIDS. Some cancers are known to have viral origin. For example, human papillomavirus causes genital warts, which are a pre-stage of cervical cancer.

All viruses carry genetic information either in the form of deoxyribonucleic acid (DNA) or ribonucleic acid (RNA), which encodes a relatively small number of genes. This genetic information is protected by a protein coat known as capsid. Some viruses have an ulterior surrounding envelope of fat. A simple virus is shown in Figure 3.1.

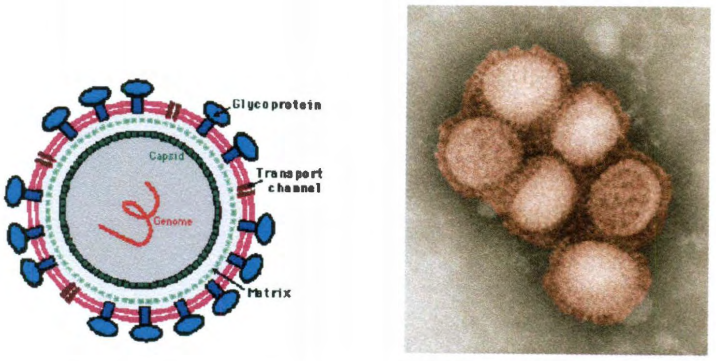


Figure 3.1 : Virus. On the left, a sample enveloped virus (figure from http://www.microbiologybytes.com/introduction/graphics/glyco.gif). On the right an electron micrograph of the Swine Flu virus (photo from http://www.digitaljournal.com/img/8/9/9/i/5/0/5/o/SwineFluCells.jpg)

3.1.2 Virus propagation cycle

Viruses lack independent metabolic activity and therefore cannot replicate by themselves. They have to rely on the basic protein synthesis machinery of a cellular host as well as host's transcription machinery. To achieve this goal, they have to infect a cell (be internalized by a cell). Once inside, the protective capsid uncoats exposing the genetic material that carries the information to replicate itself and produce various viral proteins. The transcription/translation machinery of the cell produces replicates that spontaneously reassemble as new virus progeny of the order of thousands virions per cell, which then leave the cell, often in a burst that kills it. The process is illustrated in Figure 3.2.

3.1.3 Virus morphology

Viruses differ among themselves with respect to shape, size, genetic content, and if they are enveloped or not. It is not possible to systematically classify them into a single phylogenetic tree according to their relatedness.

Among the smallest in size are the virions of parvovirus, which are less than 20nm in diameter, containing a single-stranded DNA genome with less than $5,000$ nucleotides. On the other end, the virions of poxvirus range among the largest. Their

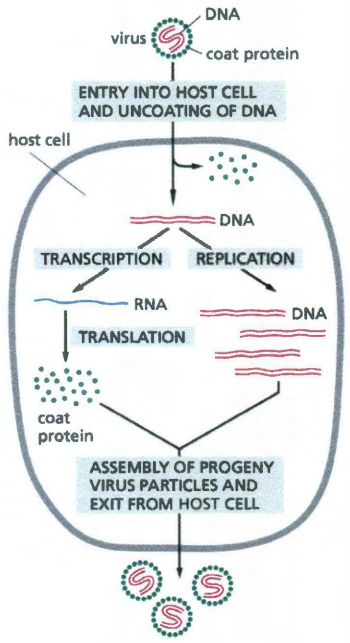


Figure 3.2 : Life cycle of a simple virus. Typical propagation cycle of a simple double-stranded DNA virus with a single coating protein (figure from Ref. [1]).

size is about 450nm, containing a double-stranded DNA genome of about 270,000 nucleotide pairs. Examples are shown in Figure 3.3.

The genome (DNA or RNA) of the smallest viruses contain just a few genes. The largest ones, with double-stranded DNA, may contain hundreds of genes. Some of the genomes are circular, while others have ad-hoc ends needed to replicate the nucleotides at the extremes of the strand. Examples are shown in Figure 3.4.

All viral genomes encode three types of proteins:

- Proteins to replicate the genome.
- Proteins for packaging the genome.
- Proteins to alter the normal functioning of the host cell and favor the replication of the virions.

Some viruses encode also another class of proteins aimed to change or repress the immune defense mechanisms of the host.

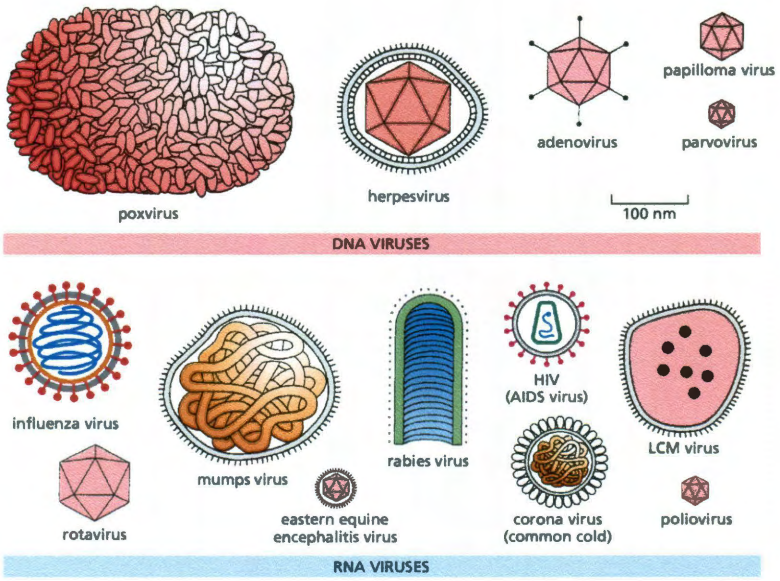


Figure 3.3 : Virus Morphology. Figure from Ref. [1].

The capsid enclosing the genome may be composed of a single or different proteins, regularly arranged in repeating patterns and layers. The module formed by the genome and the capsid is collectively known as the nucleocapsid.

Enveloped viruses have the nucleocapsid enclosed by a lipid bilayer membrane acquired by the virus when budding from the host cell plasma membrane (see figure 3.5). Viruses of this kind usually do not kill the cell as a result of infection, and often are responsible for chronic diseases. Non-enveloped viruses, on the other hand, often kill the host cell by lysing it (breaking the cell open).

3.1.4 Infection of a host cell by a virus

In order to infect a cell, a virus first needs to bind to its surface. To this end, the viral coating proteins target a specific receptor on the surface of the cell. This receptor can be a protein or not. For example, herpes simplex virus binds to heparan sulfate proteoglycans. Depending on the specificity of the virus, it can target either an abundant type of receptor, or one that is very specific to certain type of cell. Some viruses target only one kind of receptor, other several. Often virions, beyond their

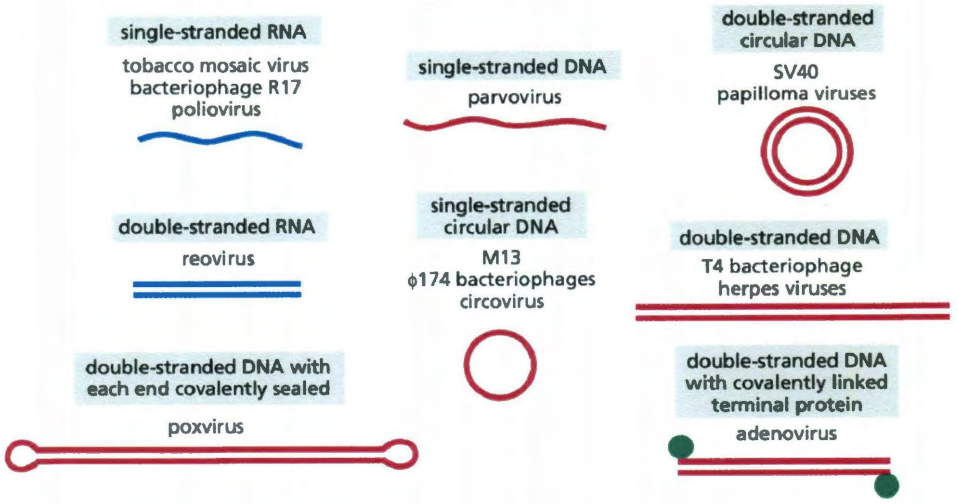


Figure 3.4 : Types of viral genomes. Figure from Ref. [1].

primary receptor, need a secondary one (a co-receptor) to attach and enter the host cells. An example of this is HIV, shown in Figure 3.6.

Virions use different strategies to enter the cell. Enveloped viruses enter the host either by fusion with the plasma membrane or, after endocytosis, i.e. the process by which cells engulf external entities with the cell membrane, with the endosomal membrane (the membrane bound compartment inside eukaryotic cells). HIV is an example of the first type, influenza virus of the second. These processes are illustrated in Figure 3.7 parts A and B. They then undergo a conformational change, releasing the genome.

The mechanisms used by non-enveloped viruses are less clear, as it is not trivial how relatively large structures are able to cross the plasma or endosome. They generally either form a pore in the endosome to inject the genome, or disrupt the endosome after endocytosis, and then release the genome. Poliovirus is an example of the first mechanism, and adenovirus of the second (see Figure 3.7 parts C and D).

Enveloped viruses need to acquire their membrane from host cell phospholipids. In the simplest cases, viral proteins are inserted into the endoplasmic reticulum (ER) membrane and follows a path through the Golgi apparatus to the plasma membrane

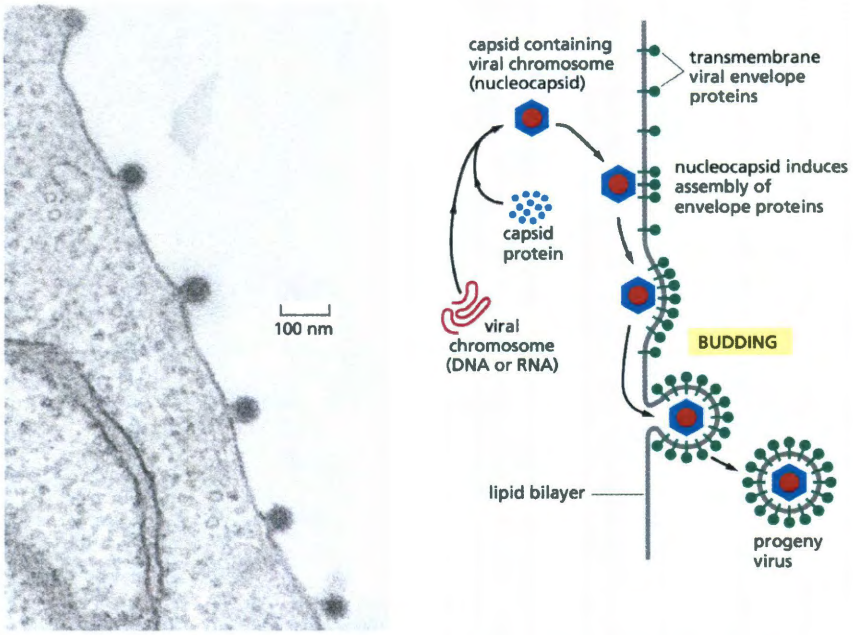


Figure 3.5 : Virus with lipid envelope. Figure from Ref. [1].

that surrounds the cell. Nucleocapsids then assemble at the plasma membrane, budding of. This is the mechanism used by HIV. For a more complicated strategy used by herpes virus, see Figure 3.8.

3.1.5 Virus evolution

Even if the complexity and specificity of the interactions between viruses and the infected cells should almost rule out random mutation, new strains are constantly emerging, and old viruses are constantly changing making it difficult to treat familiar infections. On one side, they replicate very quickly. As an example, poliovirus undergoes a 2% change in its genome in 5 days. Humans, instead, needed about 8 million year of divergent evolution to acquire a 2% genome difference compared to chimpanzees. Moreover, selective pressures such as those exerted by the host's adaptive immune system or drugs favor fast genetic change.

The replication mechanisms are among the main causes of mutation. In retroviral genomes, for example, point mutations are acquired during the replication cycle as

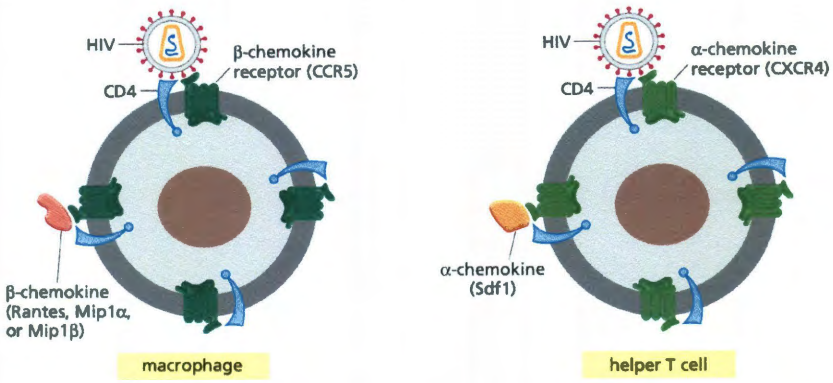


Figure 3.6 : HIV virus interacting with receptor and co-receptor. Figure from Ref. [1].

the viral transcriptase that produces DNA from the viral RNA is unable to correct nucleotide misreadings. The high mutation rate is in some way beneficial for the virus, as the process of mutation and selection within each host produces variants best adapted and resistant to immune responses and drugs.

Only about one third of the viral genome is invariant, while nucleotide sequences in certain parts of the genome may differ by about 30%, making it very difficult to develop vaccines for certain types of viruses, such us HIV.

3.1.6 The innate immune system

Innate immune responses, different from the adaptive immune responses, are not specific to a particular pathogen. Among the different kinds of innate immune responses, one that effectively combats virus infections is the ability of host cells to aggressively degrade double-stranded RNA, a common intermediate in viral replication [2], as well as any single-stranded RNA sharing sequence identity with the associated double-stranded trigger. Another kind is related to specialized sets of proteins that become activated once they recognize conserved features of pathogens, or pathogen-associated molecular patterns (PAMPs), starting a cascade of reactions aiming to stop the infection.

Many of the mammalian pattern recognition receptors that trigger innate im-

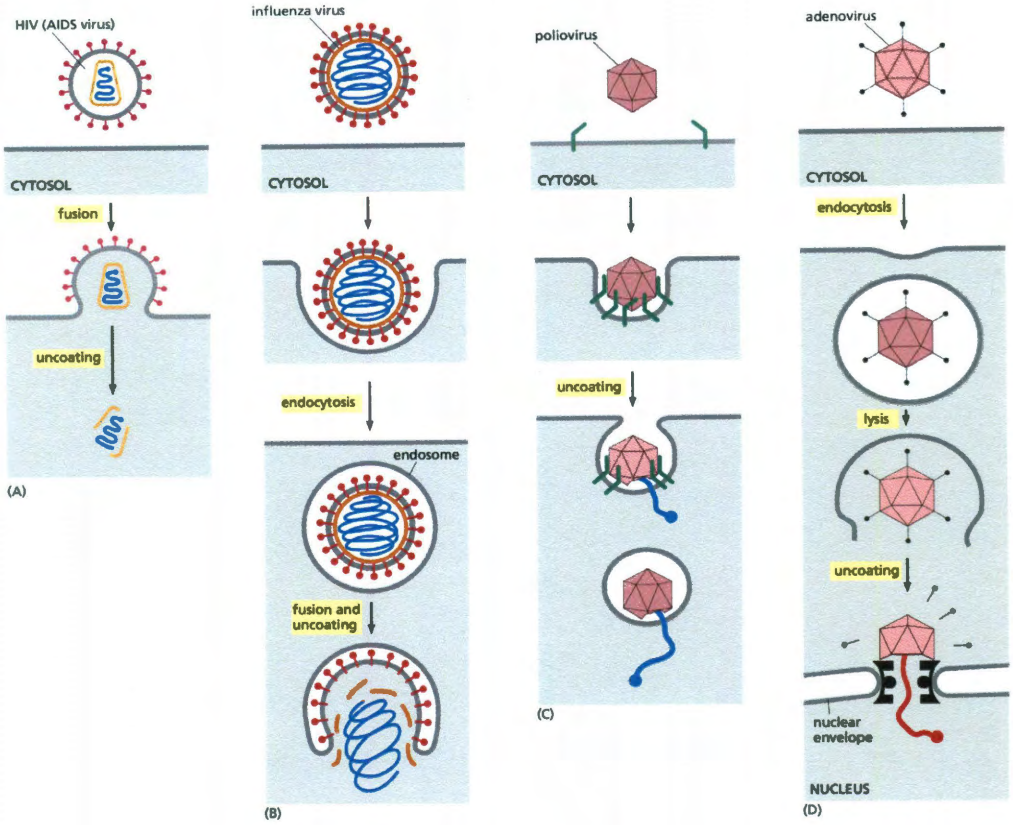


Figure 3.7 : Virus uncoating. Figure from Ref. [1]

mune responses belong to the family of Toll-like receptors (TLR). Pathogens produce immunostimulants that are recognized by TLRs. Most of them are on the cellular surface of macrophages, dendritic cells and certain epitheliar cells, but some of them are associated with intracellular membranes.

As host cell ribosomes translate viral proteins and host cell membranes form the membranes of enveloped viruses, the only general way that the host cell has to recognize a virus is through particular motifs of the viral genome.

3.1.7 Interferons (IFNs)

Double-stranded RNA (dsRNA) induces production and secretion of two cytokines: interferon-α (IFNα) and interferon-β (IFNβ), which act in both an autocrine (directed towards the infected cell) and a paracrine fashion (directed towards uninfected

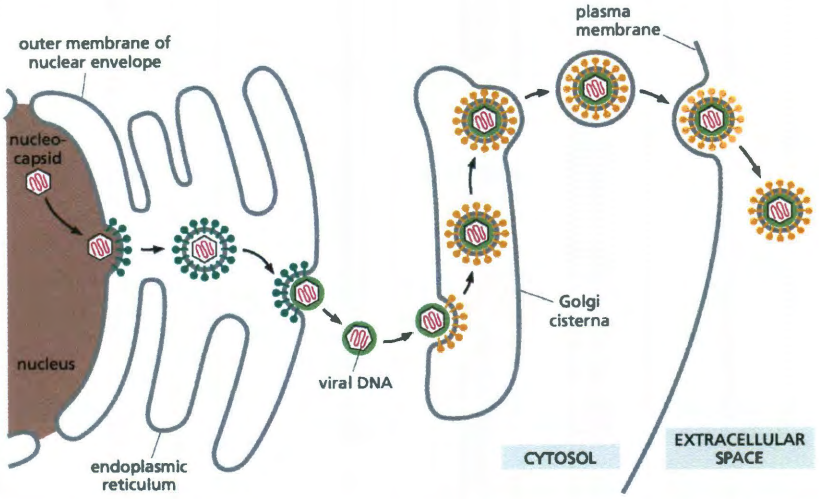


Figure 3.8 : Viral envelope acquisition. Herpes virus nucleocapsids assemble in the nucleus. They then bud through the inner nuclear membrane into ER, acquiring a membrane coat. The virus particles then lose this coat while fusing with the outer nuclear membrane to escape into the cytosol. Then, the nucleocapsids bud into the Golgi apparatus and bud out again on the other side, acquiring two new membrane coats. The external is lost when budding from the cell (figure from [1]).

neighbors). Interferons bind to their cell-surface receptors stimulating transcription of specific genes through the JAK/STAT intracellular signaling pathway [62] (see figure 3.9). This activates more than 300 genes, including many cytokines. Interferon response seems to be a non-specific reaction of mammalian cells to a viral infection, and different viral components (not only dsRNA) can initiate it.

Beyond the effects on host cell gene transcription, interferons activate a latent ribonuclease, which degrades ssRNA in a non-specific way. Moreover, they indirectly activate a protein kinase that inactivates via phosphorylation the protein synthesis initiation factor eIF-2 [1]. This shuts down most protein synthesis in the infected cell.

Then, by degrading most of its RNA and temporarily stopping most of its protein synthesis, the infected cell inhibits viral replication without killing itself.

However, if this does not suffice, the cell kills itself by apoptosis, often through the action of a killer lymphocyte.

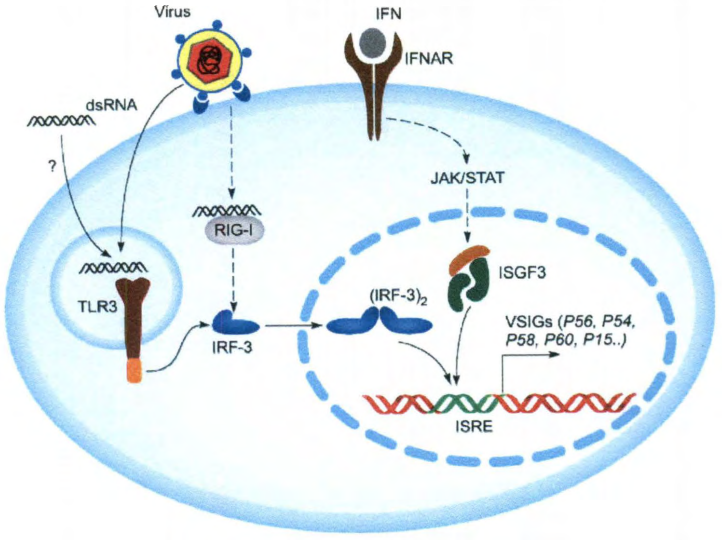


Figure 3.9 : TLR3, IFNs and dsRNA. Viral dsRNA can be endocytosed and start TLR3 mediated signaling cascade activating IRF-3. Cytoplasmic dsRNA, by a TLR3 independent pathway using RIG-I, also activates IRF-3. Type I IFNs start JAK/STAT signaling pathway by binding to IFN receptor (IFNAR) and activating ISGF3G. Both IRF-3 and ISGF3G bind to the same cis-element, ISRE, in the promoter regions of viral stress-inducible genes (VSIGs) (figure from Ref [62]).

3.2 Nuclear factor kappa-light-chain-enhancer of activated B cells (NF-κB)

Nuclear factor kappa-light-chain-enhancer of activated B cells (NF-κB), whose most abundant form is the heterodimer p50/p65 (NF-κB1/RelA), is a gene regulatory protein, or a transcription factor. NF-κB is a key player in many stressful, inflammatory, and innate immune responses. In unstimulated cells, NF-κB is in a dormant state, sequestered in the cytoplasm by a family of inhibitors, mainly by the Inhibitor of κB α (IκBα).

As shown in the simplified schematic of Figure 3.10, when an external stimuli (such as Tumor Necrosis Factor α, or TNFα) activates specific receptors, the IκB Kinase (IKK) becomes activated, subsequently phosphorylating IκBα and leading to its degradation. The freed NF-κB then enters the nucleus, activating a number of target genes. Among them, at least two are important to regulate NF-κB itself: A20

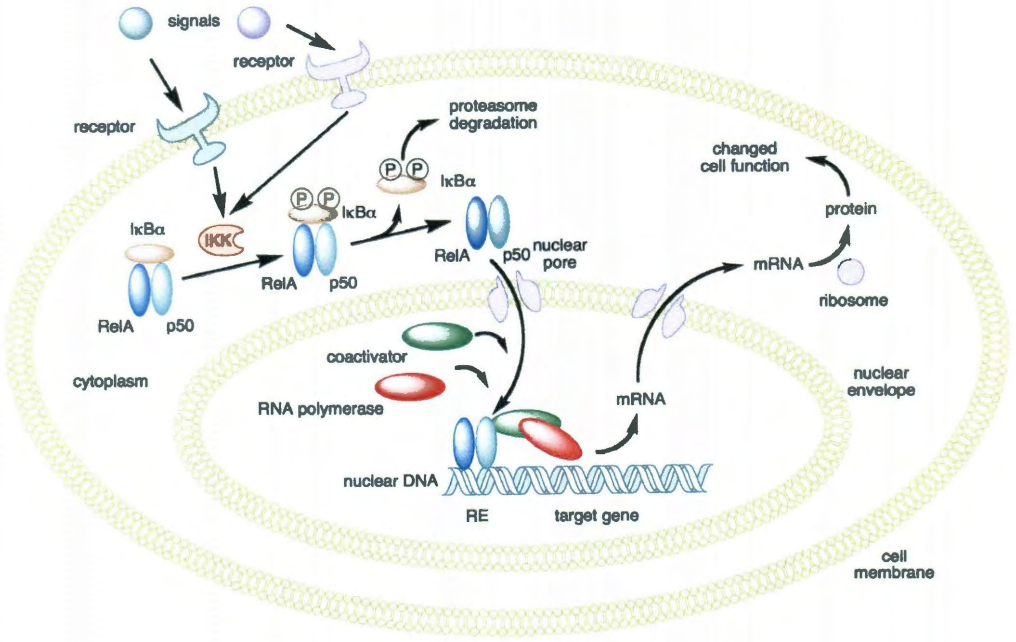


Figure 3.10 : Simplified mechanism of NF-κB action. Image from http://upload.wikimedia.org/wikipedia/commons/5/5e/NFKB_mechanism_of_action.png

and IκBα. A20 inactivates IKK, and IκBα binds NF-κB inhibiting its action. NF-κB is one of the key activators of INFβ gene, as shown in Figure 3.11.

3.3 Interferon regulatory factor 3 (IRF-3)

IFN regulatory factor 3 (IRF-3) belongs to the family of nine human IRFs that have been identified so far (IRF-1 to IRF-9). IRF-3 is a key component in the induction of type I interferons after viral infection or double stranded RNA induction, and it is constituted by 427 amino acids. Transcriptional activation is mediated by the IKK-related kinases TBK-1 and IKKϵ, whose C-terminal phosphorylation induces a conformational change in IRF-3 that allows dimerization and binding to IRF-7 and nuclear translocation [66]. Inactive IRF-3 constitutively translocates into and out of the nucleus, while phosphorylated IRF-3 is retained in the nucleus where it induces transcription of IFNβ and other genes [30]. IRF-3 role in signaling pathway

is summarized in Figures 3.11.

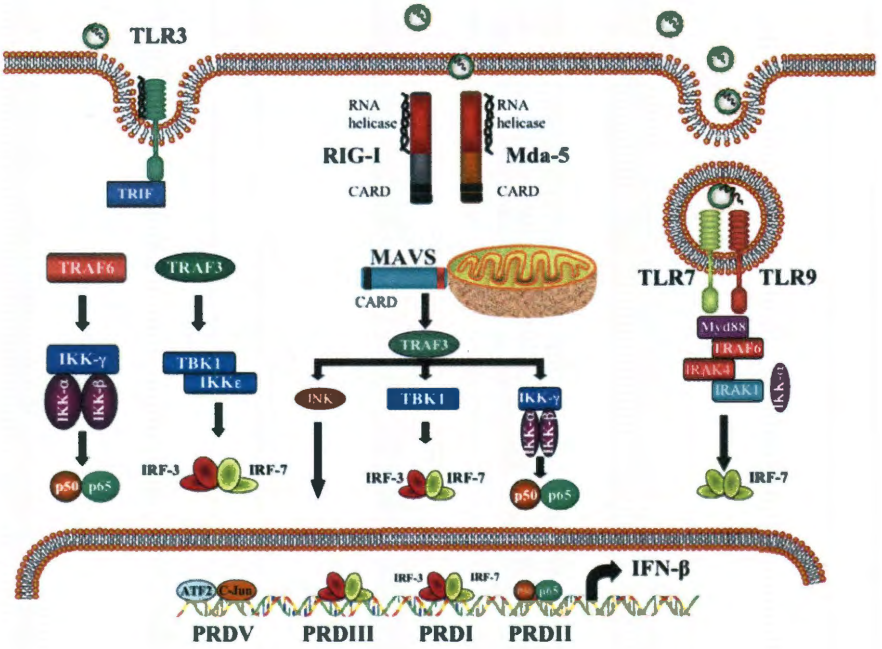


Figure 3.11 : Signaling pathways that recognize virus infection. Figure from Ref [30].

3.4 RIG-I like receptors and Toll-like receptors

3.4.1 RIG-I like receptors

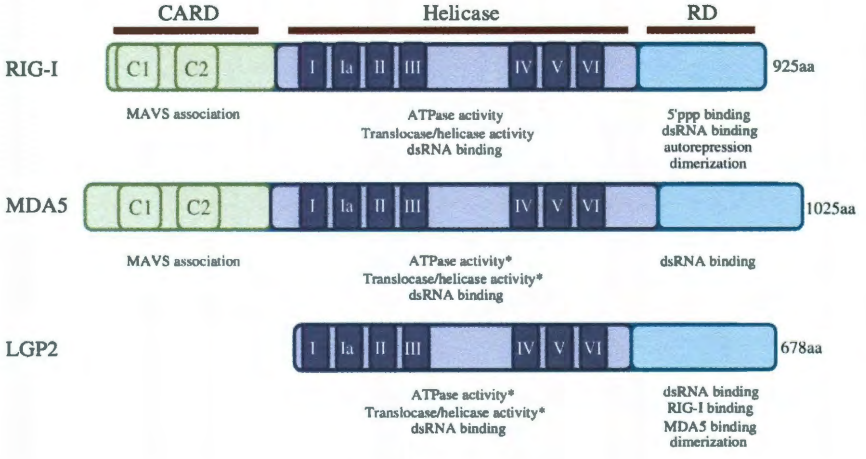


Figure 3.12 : RLR family of cytoplasmic proteins. Figure from Ref [4].

RIG-I like receptors (RLR) are a family of cytoplasmic viral sensors, composed

by three members [4]:

- Retinoic acid-inducible Gene 1 (RIG-I), also known as Ddx58.
- Melanoma-differentiation-associated gene 5 (MDA5), also known as Ifih1 or Helicard.
- Laboratory of Genetics and Physiology 2 (LGP2).

Both RIG-I and MDA5 contain a typical ATP-dependent helicase domain. The N-terminus of these proteins encodes two caspase recruitment domains (CARDs) RIG-I recognizes dsRNA with 5ppp U-rich 5ppp containing RNA, sensing the majority or RNA viruses, while MDA5 recognizes long dsRNA (over 2 kb), sensing the majority of picornaviruses. LGP2 lacks the card domain [4]. We will concentrate in the signaling pathway driven by RIG-I.

3.4.2 Toll-like receptors

Toll-like receptors (TLRs) are a family composed, up-to date, of 13 mammalian members [4]. These receptors are constituted by an extracellular leucine-rich repeat domain which recognized pathogens, and a cytoplasmic domain with homology to IL-1R which is involved in downstream signaling [4]. TLRs function as pattern-recognition receptors sensing PAMPs found in pathogens [36]. We will concentrate in TLR3, being the more specific among TLRs in sensing dsRNA.

3.4.3 RIG-I, Mitochondrial anti-viral signaling protein (MAVS), and TNF receptor-associated factor 3 (TRAF3)

Double- and single-stranded viral RNAs bind to RIG-I [34, 35, 40] inducing its ubiquitylation [17, 52]. This complex then binds to Mitochondrial Anti-Viral Signaling protein (MAVS; also known as IPS-1, VISA, and CARDIF) [3]. This oligomer, known as RIG-I-MAVS, recruits TNF Receptor-Associated Factors (TRAFs) 2, 3 and 6 [38].

This resulting complex activates the complex TANK-binding kinase 1 (TBK1) and Inhibitor of κB kinase ϵ (IKKϵ) [23,71], which in turn activates IRF3 [16,63], that induces the production of type I interferons (IFNs) [41]. Figure 3.13 depicts a schematics of this pathway.

3.4.4 Toll-like receptor 3, TIR-domain-containing adapter-inducing interferon-β (TRIF), and TNF receptor-associated factor 6 (TRAF6)

dsRNA activates TLR3 [2, 62], which binds to the adaptor molecule TIR-domain-containing adapter-inducing interferon-β (TRIF) [70] and to TRAF6 [62], phosphorylating IκBα via IKKα-IKKβ-IKKγ [62], which frees NF-κB that translocates into the nucleus [31,65]. See Figure 3.14 for a schematics of this pathway.

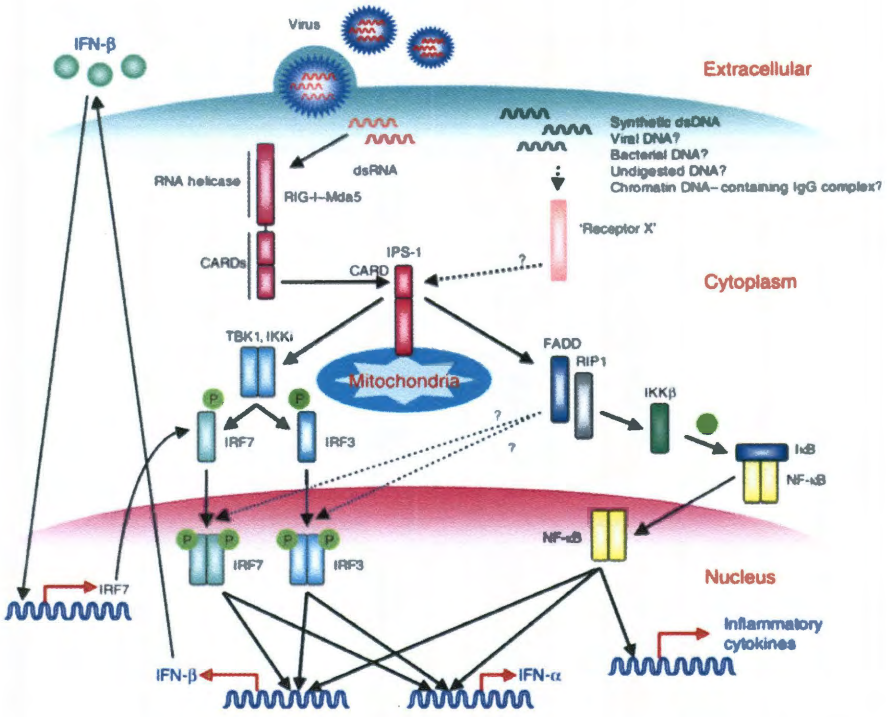


Figure 3.13 : RIG-I-MAVS signaling pathway. Figure from Ref [36].

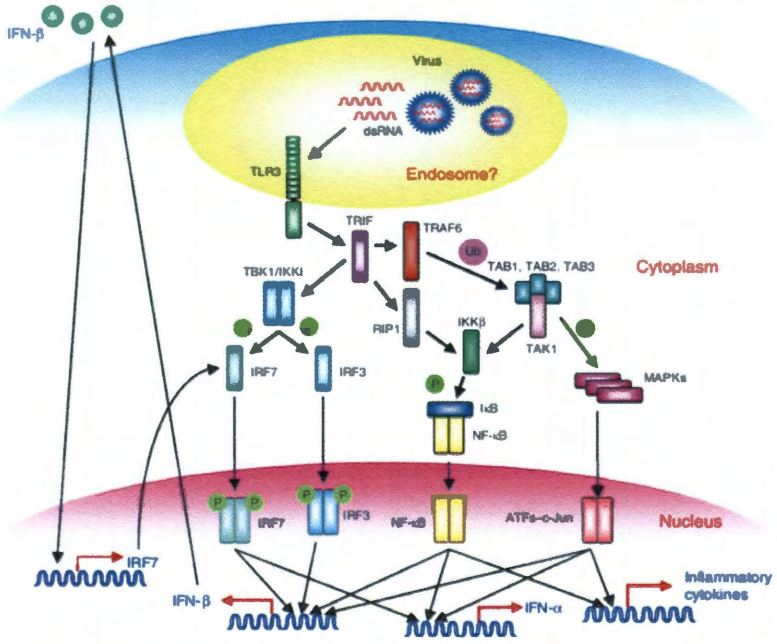


Figure 3.14 : TLR3-TRIF-TRAF6 signaling pathway. Figure from Ref [36].

Chapter 4

Comparison between deterministic and stochastic effects using bioPN

In this chapter we introduce the software library bioPN (biological Petri Nets) and its applications. It consists of three parts. In part one we describe bioPN. In part two we validate numerical methods used in bioPN against analytical solutions. In part three we compare deterministic and stochastic solutions for different simple pathways, and we show a case where bi-stability arises. This latter was published paper about crosstalk between the transcription factors NF-κB and p53 [58].

4.1 bioPN

bioPN is a R library of C functions that can be used to simulate time-dependent evolution of biochemical reactions.

The model is defined as a place/transition Petri Net (defined in Chapter 2). It can be either deterministically solved using an explicit Runge Kutta Dormand Prince 45 method, simulated using four highly optimized variants of the stochastic simulation algorithm, or as a deterministic/stochastic hybrid, according to the Haseltine and Rawlings' algorithm, or using the Partitioned Leaping Algorithm (PLA).

The library has been optimized for speed and flexibility, allowing to choose among several simulation methods with a single model definition.

4.1.1 Background

Analytical solutions to most but the basic stochastic systems of biochemical reactions in cells are intractable, making necessary the use of computers to perform runs of stochastic simulations. For better approximations, high numbers of runs are needed,

so that optimized methods are necessary to perform the task in an acceptable time. Many approaches to perform exact simulations have been proposed. In the case of eukaryotic cells, the number of reactions of protein production and degradation is so high that exact methods, such as the stochastic simulation algorithm (SSA) [19] may become impractical. Several approximations have been developed to speed-up algorithms (for review see Chapter 8 of [69]).

Of the software available, STOCKS [37] is a non-optimized C++ implementation of the Gillespie SSA method. COPASI [32] is a multi-platform C++ application that allows pure stochastic and pure deterministic simulation. The software also imports SBML files. StochKit [39] is a C++ application that includes the SSA algorithm and approximate stochastic methods. The model is defined as a Systems Biology Markup Language (SBML) file. GillespieSSA [56] is a package programmed in R capable of performing direct and approximate stochastic methods. The user defines structures for the constants and the stoichiometry matrix, and provides expressions for reaction propensities.

In this chapter, we describe and discuss a new software tool, bioPN, which we developed to optimize the computation speed without compromising accuracy. The user of bioPN is offered the choice of several methods, ranging from variants of SSA to various approximate algorithms and their combinations.

4.1.2 Approach

As a means of representing the biochemical networks, we chose the paradigm of Petri Nets (reviewed in 2.2) as it is closely related to the way reactions are defined.

R [59] was chosen as the backbone because it is a robust language for statistical data analysis that allows post-processing of the results and has plotting capabilities. It is licensed under the GPL, available free of cost, runs on several architectures (including Unix, Linux, Windows, and Mac), and has an active and growing developer

and user community.

R is an interpreted language, which makes it slow compared to compiled ones. However, this limitation does not apply in our case, as the bioPN functions run entirely in a compiled C code. Moreover, R provides a C library that can be called from compiled code. Uniform random numbers are generated by the R library functions unif_rand() and rexp(), which have passed strict tests for randomness. BioPN allows the user to choose a specific generator, or use R's default, currently the Mersenne-Twister [48].

In the deterministic and hybrid mode [27] fast reactions are approximated with an explicit Runge Kutta Dormand Prince 45 method [11]. The Partitioned Leaping Algorithm (PLA) [24] dynamically divides the reactions into four categories. Very slow reactions are simulated using Next Reaction method, slow ones with the τ-leaping algorithm, medium ones with chemical Langevin equation, and fast ones deterministically (Euler method).

To explain the principles of bioPN, consider the following set of chemical reactions:

$$\text{Gi} \xrightarrow{c} \text{Ga} \tag{4.1}$$

$$\text{Ga} \xrightarrow{b} \text{Gi} \tag{4.2}$$

$$\text{Ga} \xrightarrow{H} \text{Ga} + \text{mRNA} \tag{4.3}$$

$$\text{mRNA} \xrightarrow{K} \text{mRNA} + \text{Protein} \tag{4.4}$$

$$\text{mRNA} \xrightarrow{m} \otimes \tag{4.5}$$

$$\text{Protein} \xrightarrow{r} \otimes \tag{4.6}$$

This model is composed by 4 places (reactants, products) and 6 transitions (reactions) that define this model. The places are: the inactive gene (Gi), the active gene (Ga), the mRNA and the Protein. The transitions are: gene activation, gene inactivation, transcription, mRNA degradation, translation, protein degradation.

More generally, consider p places and t transitions. To parametrize it as a Petri Net, we need to define two $t \times p$ matrices: *pre* will have in each row (for each transition) the stoichiometry of the reactants (left side of the reaction), while *post* the stoichiometry of the products (right side of the reaction).

Another needed structure is h, a list with t elements, which is used to identify reaction propensities. For flexibility, each of its elements can be either a number, an R function, or a pointer to a C function. If it is a number, it is considered a rate constant and the corresponding mass-action kinetic propensity is calculated on the fly according to the order of the reaction and its stochastic or deterministic nature. The other two options are used for more complicated or ad-hoc situations, such as negative feed-backs. In these cases the functions supplied are called with the current time and marking (state) of the system as arguments, and the returned value is used as the propensity.

The $p \times 1$ vector M (marking) is used to define the initial conditions. For our example, and assigning values to the rate constants and initial conditions (the gene is originally inactive), the model R code reads as follows:

```
model <- list(
    pre <- matrix(c(1,0,0,0, 0,1,0,0, 0,1,0,0,
                    0,0,1,0, 0,0,1,0, 0,0,0,1),
                  ncol=4,byrow=T),
    post <- matrix(c(0,1,0,0, 1,0,0,0, 0,1,1,0,
                     0,0,0,0, 0,0,1,1, 0,0,0,0),
                   ncol=4,byrow=T),
    h <- list(c=3, b=2, H=10, m=1, K=6, r=0.25),
    M <- c(1,0,0,0))
```

This model definition can be used to simulate the system by different methods. For a deterministic solution, the library automatically creates the system of ordinary differential equations (ODEs). The call to the function has the form:

```
Sim <- RungeKuttaDormandPrince45(model, T = 200, delta = 1)
```

which simulates the system for 200 time units (T = 200), saving the results each 1 time unit (delta = 1).

For an exact stochastic run, four methods can be invoked. For example, for an

optimized version of Gillespie SSA, the call has the form:

```
1 set.seed(19761111)
2 Sim <- GillespieOptimDirect(model, T = 200, delta = 1, runs = 1000)
```

A series of exact stochastic simulations can be performed in one call to the function (runs = 1000), so the optimization of the data structures used by the algorithms is performed only once. The exact stochastic simulation algorithms included are:

- GillespieOptimDirect, which reorders reactions so the ones occurring more often are moved to the top minimizing the linear search to find the "reaction that occurred".
- GillespieDirectGB, that optimizes the SSA according to Gibson and Bruck [18].
- GibsonBruck, the next reaction method by Gibson and Bruck [18].
- GillespieDirectCR, that applies the constant-time composition/rejection algorithm [64].

The different algorithms perform better in different situations. The composition/rejection algorithm, for example, excels when the number of reactions is very large [64].

We do not include pure approximate algorithms, such as τ-leaping or chemical Langevin equations, as we have not found any cases that would justify their use. Their implementation, if needed, would be straightforward. Both approaches are implemented as a part of the partitioned leaping algorithm (PLA) [24], which dynamically partitions the transitions into four groups, according to their propensities. Very slow reactions are simulated using the next reaction method, slow ones using the τ-leaping algorithm, medium ones using the chemical Langevin equation, and fast ones deterministically. The corresponding function is PartitionedLeaping.

In most biological systems there are reactions that occur very often (*fast* reactions), while others occur rarely (*slow* reactions). Haseltine and Rawlings [27] proposed to use a hybrid algorithm that simulates slow reactions using SSA, and the fast

reactions either deterministically or using the chemical Langevin equations (CLE) algorithm. In this case, an extra parameter (slow) is used to determine which reactions are considered slow and fast. If, for the previous example, gene activation and deactivation are considered slow reactions, whereas mRNA and protein production and degradation are considered fast, we set model$slow <- c(1,1,0,0,0,0).The call to the function in this case has the form:

```
1 Sim <- HaseltineRawlings(model, T = 200, delta = 1, runs = 10000)
```

To account for the cases with times at which the behavior of the system changes, T can be assigned an R function or a pointer to a C function. This function receives the current state and time of the system and modifies the state. As an example, one of the places can be a yes/no flag that allows some reactions to happen during a section of the run, but disallows them in other sections.

4.1.3 Discussion

The definition of the model as a Petri Net is convenient as it is closely related to the way reactions are defined. However, it has the side effect of generating very sparse matrices. To account for this, the *pre* and *post* matrices are analyzed before the first run, and the relevant information is stored in internal non sparse structures, to avoid unnecessary computations, such as multiplications by zero followed by summation of the zeros. Other structures are aimed at speeding the algorithm, such as identifying the propensities to be recalculated after the last transition occurred or propensities affected by the deterministic evolution in hybrid models.

The matrix formulation of this paradigm can be advantageously used to perform exploratory search of models using mass-action laws, by automatically modifying the coefficients of both *pre* and *post* matrices before calling the functions. Fitting criteria could then be applied to the results to create a ranking of successful models.

R has functions to import SBML files. If needed, methods could be added to

bioPN to perform the translation from SBML to Petri Net structures.

The differential equation solver, an explicit Runge Kutta Dormand Prince 45 method, is a very accurate integrator with adaptive step size selection. A parameter controls the level of accuracy, with a default of $1e-9$. Reducing the accuracy produces acceleration, with negligible loss of precision. For hybrid models, time-dependent propensity is computed using a trapezoidal approximation that is fast and accurate for the required purposes.

In principle, PLA should represent the best compromise between precision and performance. However, the repeated re-partitioning and step selection, and rejections of steps in which negative number of species are generated, can prove computationally costly in some situations. The method of Haseltine and Rawlings [27] appears to be most successful, in the sense of an acceptable loss of accuracy compared to a great gain of performance.

4.2 Validation of bioPN against derived analytical expressions

The model defined by Equations 4.1 to 4.6 was studied by Paszek [53], who derived steady state analytical expressions for the mean and variance of mRNA and protein for the stochastic problem and two hybrid variants of stochastic and deterministic treatment of the reactions.

Our goal is to compare the results from analytical expressions with simulations performed with bioPN to verify the correctness of the implemented algorithms.

We will use the parameter values suggested by Paszek for procaryotic cells: $c = 3$, $b = 2$, $H = 10$, $K = 6$, $m = 1$, and $r = 0.25$.

4.2.1 Deterministic integration of system of ODEs

Analytical expressions for the expected values of mRNA and Protein are given by [53]:

$$\mathrm{E}\left[\mathrm{mRNA}\right] = \frac{c}{c+b}H \tag{4.7}$$

$$\mathrm{E}\left[\mathrm{Protein}\right] = \frac{K}{r}\mathrm{E}\left[\mathrm{mRNA}\right] \tag{4.8}$$

These expressions are not only valid in the deterministic case, but also for the stochastic and any deterministic/stochastic hybrid case. Expressions for the variance, however, will be different for each situation.

Using the parameters defined above, the results are: $\mathrm{E}\left[\mathrm{mRNA}\right] = 6$ and $\mathrm{E}\left[\mathrm{Protein}\right] = 144$. These values (green lines) are compared with the result of performing the deterministic integration with the bioPN function RungeKuttaDormandPrince45 (blue lines) and shown in Figure 4.1. The results agree validating the deterministic Runge Kutta Dormand Prince 45 integration algorithm programmed.

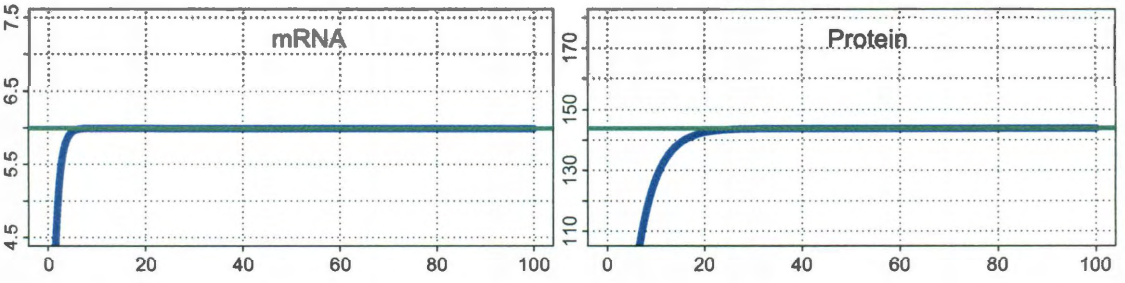


Figure 4.1 : Deterministic system. Horizontal axis: time units. Vertical axis: number of units. Green lines: Analytical results. Blue lines: Deterministic integrations using bioPN.

4.2.2 Completely stochastic simulations

Expressions for the variances of mRNA and Protein when all the reactions are considered stochastic are given by [53]:

$$\mathrm{Var}\left[\mathrm{mRNA}\right] = \frac{b}{c(1+c+b)}\mathrm{E}^2\left[\mathrm{mRNA}\right] + \mathrm{E}\left[\mathrm{mRNA}\right] \tag{4.9}$$

$$\begin{aligned}\text{Var}\left[\text{Protein}\right] =& rb\frac{1+c+b+r}{c(1+r)(1+c+b)(r+c+b)}\text{E}^2\left[\text{Protein}\right]+ \\ & \frac{r}{1+r}\frac{\text{E}^2\left[\text{Protein}\right]}{\text{E}\left[\text{mRNA}\right]}+\text{E}\left[\text{protein}\right]\end{aligned} \tag{4.10}$$

This case was used to validate the four completely stochastic functions in bioPN (GillespieOptimDirect, GibsonBruck, GillespieDirectGB, GillespieDirectCR), which showed conformance with analytical results. Results for means and variances for 1000 runs for GillespieOptimDirect are shown in Figure 4.2.

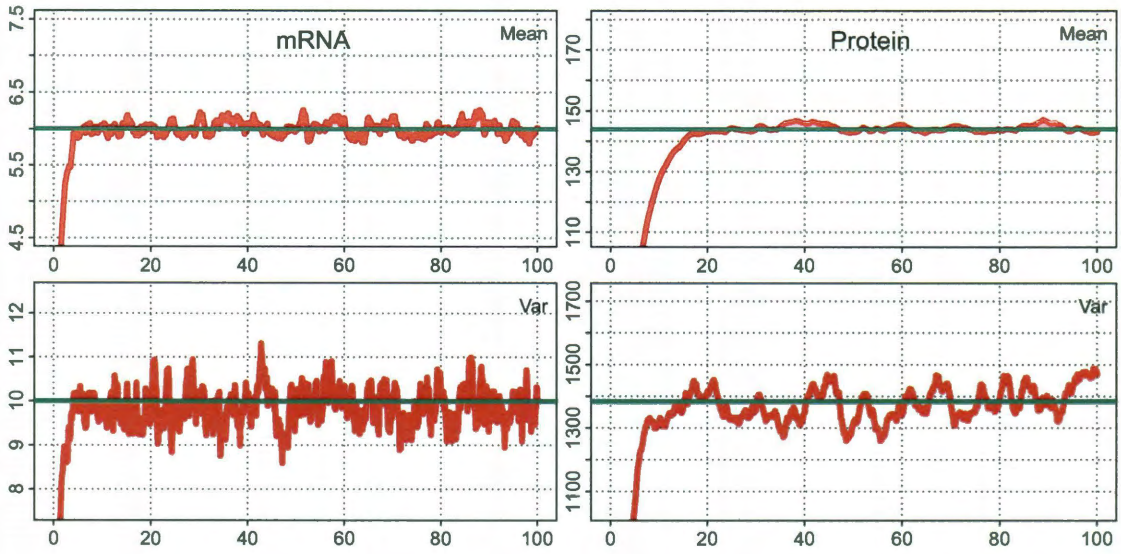


Figure 4.2 : All reactions stochastic. Horizontal axis: time units. Vertical axis: number of units. Green lines: Analytical results. Red lines: Simulations using bioPN (1000 runs).

Comparisons for hybrid algorithms were also performed showing similar concordance to analytical results, and equations and plots are presented in Appendix A, starting on page A.2.1. Results for extended series of 10,000 runs are also presented in Appendix A, starting from page 164.

4.3 Comparison between deterministic and stochastic systems

In this section we will compare deterministic integration (i.e. solving a system of ordinary differential equations) against averages of stochastic runs, for the solution of time evolution of gene regulatory networks.

We will study the following systems:

- Model by Paszek [53] used in Section 4.2 to validate bioPN against steady state analytical expressions of mean and variance.
- Model by Paulsson et al. [54], used to introduce the key concept of "stochastic focusing".
- Expansion of Paszek's model by introduction of gene repression (negative feedback) by the produced protein.
- Modification of this last model by replacing the repressing protein by its dimer. This final model will coincide with the one used to illustrate Petri nets in Chapter 2.

4.3.1 Unregulated gene

We will start by comparing deterministic and stochastic methods for the model described by Equations 4.1 to 4.6. This illustrates the "central dogma of molecular biology". The chemical species considered are a given gene and the corresponding mRNA and protein molecules. The gene becomes activated with constant rate and inactivated with a rate depending only on the state of the gene itself. As the gene state does not depend on the presence of other molecules (mainly proteins), it is called an unregulated gene. This gene transcribes mRNA, which translates a protein. Both mRNA and protein degrade. The rate constants used are the same as the ones used to validate bioPN ($c = 3$, $b = 2$, $H = 10$, $K = 6$, $m = 1$, and $r = 0.25$).

Results for active gene (Ga), mRNA, and protein are presented in Figure 4.3 where it is represented:

- in blue: the time evolution of the solution of the system of ODEs (deterministic integration),
- in orange: the time evolution of a given stochastic run,
- in red: the time evolution of the average of 1000 stochastic runs with the corresponding pointwise 99.73% confidence interval (average $\pm$ 3 standard error).

These colors will have the same meaning for all the plots in this section.

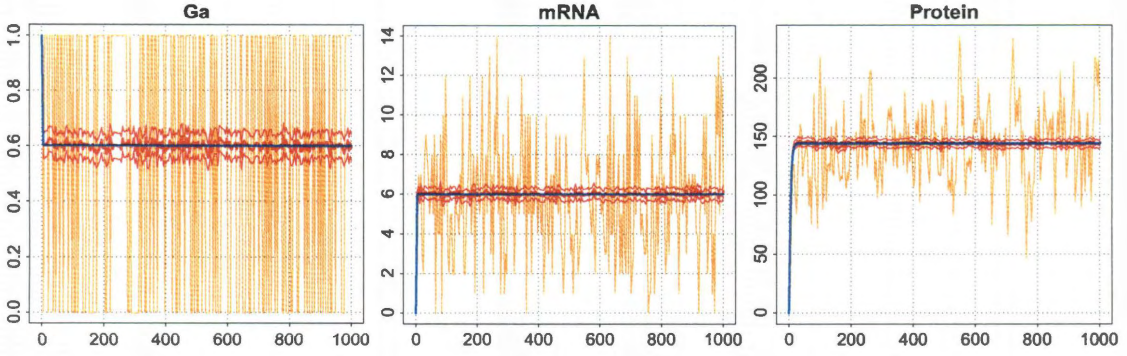


Figure 4.3 : Time evolution of mRNA and protein for unregulated gene. Horizontal axis is time units. Vertical axis is number of molecules.

The three panels of the figure show an excellent concordance between the deterministic and stochastic results, from which we can conclude that, for this case, the use of a system of ODEs to explain the evolution of the mean of the system is justified. This verifies what we have already discussed in Subsection 2.3.4 on page 17.

However, this agreement will fail soon, once we introduce a negative feedback to the gene. To provide justification to this fact, we will illustrate in the next subsection the case from literature that first pointed out (to our knowledge) a discrepancy between both approaches.

4.3.2 "Stochastic focusing" and its relation to Jensen's inequality

Paulsson et al. [54] studied the effect of signal amplification in a stochastic system, when compared to the corresponding deterministic system. The example they used was a Michaelis-Menten problem expressed by the chemical reactions:

$$\otimes \xrightarrow{k} I \tag{4.11}$$

$$S + I \xrightarrow{k_a} S \tag{4.12}$$

$$I \xrightarrow{k_p} P \tag{4.13}$$

$$P \xrightarrow{1} \otimes \tag{4.14}$$

$$\otimes \xrightarrow{k_s} S \tag{4.15}$$

$$S \xrightarrow{k_d} \otimes \tag{4.16}$$

Random variables I, P, and S are not independent in a statistical sense, but from the construction of the problem it is evident that the evolution of S is unrelated to the values that I and P at all times. In fact, the reactions that modify the values of S are expressed in Equations 4.15 (a zero order reaction), and 4.16 (a first order reaction only depending on S). The contrary is not true. Both I and P are strongly dependent on the values of S. I directly, as expressed by Equation 4.12, and P that is an amplified version of I.

When the number, per unit of time, of events corresponding to reactions expressed in Equations 4.12 and 4.13 are sufficiently high, the resulting number of molecules of I is very low and the number of molecules of S does not change in a significant way for the life span of a specific molecule of I [54]. Then Equations 4.11 to 4.14 can be replaced with the following simplified scheme:

$$\otimes \xrightarrow{kq} P \tag{4.17}$$

$$P \xrightarrow{1} \otimes \tag{4.18}$$

where q is given [54] by:

$$q = \frac{1}{1 + S/K} \tag{4.19}$$

where

$$K = \frac{k_p}{k_a} \tag{4.20}$$

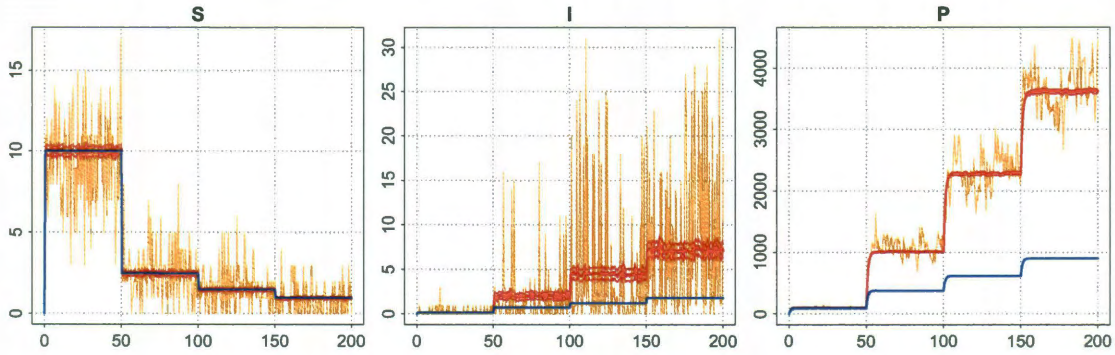


Figure 4.4 : Stochastic focusing. Time evolution of S, I and P. Horizontal axis is time units. Vertical axis is number of molecules.

We studied the system given by Equations 4.11 to 4.16 for the following rate constant values: $k = 10,000$, $k_a = k/2$, $k_p = 500$, $k_d = 100$. These parameters were kept unaltered for all time points ($0 \leq \text{time} \leq 150$) of both the deterministic integration and of the series of stochastic simulations (100 runs performed). The reaction constant of Equation 4.15 changed according to the following law:

- $k_s = 10k_d$, for $(0 \leq \text{time} < 50)$
- $k_s = 2.5k_d$, for $(50 \leq \text{time} < 100)$
- $k_s = 1.5k_d$, for $(100 \leq \text{time} < 150)$
- $k_s = k_d$, for $(150 \leq \text{time} \leq 200)$

We performed both the deterministic and stochastic studies using bioPN. The results of the evolutions of the three species are presented. The left panel of Figure 4.4

shows the time evolution of S. In this case, we can conclude that the deterministic integration has a good concordance with the averages of the stochastic realizations.

However, this is not the case for I, and even less for P, shown in the second and third panel of Figure 4.4, respectively. In these cases the concordance between the deterministic integration and the corresponding averages of the stochastic runs is poor. Using the deterministic approach to draw conclusion about the running stochastic process in a system with these characteristics is an inadequate approach.

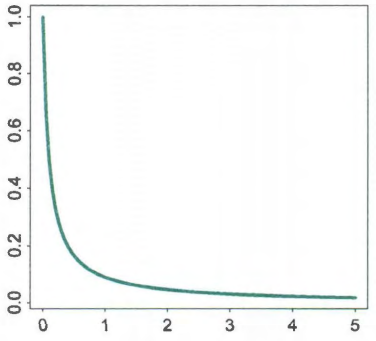


Figure 4.5 : Convexity of function q.

Jensen inequality provides a justification of the observed behavior: if X is a random variable and φ is a convex function, then:

$$\varphi(E[X]) \leq E[\varphi(X)] \tag{4.21}$$

Function q of Equation 4.19 is convex, as it can be verified in Figure 4.5. As a consequence, the deterministic integration ($\varphi(E[X])$), cannot be greater than the average of the stochastic runs ($E[\varphi(X)]$). Paulsson et al. [54] called this signal amplification effect "stochastic focusing".

If the function φ is concave, the opposite inequality holds. In this case, instead of an amplification effect, a damping effect will be in place. We show this kind of behavior in Section 4.3.4, and examples with these characteristics will also appear in further chapters concerning reaction-diffusion systems applied to early carcinogenesis

and spread of viral infection (Chapters 8 and 9), respectively.

A final question stays unanswered. Why is that, in the case of S, the deterministic integration agrees with the average of the stochastic runs? As mentioned before, the reaction propensities affecting S are: a constant, and a linear function. As both cases are straight lines, they are neither convex nor concave, then the equality replaces Jensen inequality and both results agree.

As a consequence it is possible that, in a given studied system, any possible combination of the different effects (agreement, amplification, dampening) may be coexisting simultaneously for different species.

4.3.3 Gene repressed by its produced protein

In this section we introduce the following modifications to our base model (Equations 4.1 to 4.6): Equations 4.1 and 4.2 are replaced by

$$\text{Gi} \xrightarrow{c} \text{Ga} + \text{Protein} \tag{4.22}$$

$$\text{Protein} + \text{Ga} \xrightarrow{b} \text{Gi} \tag{4.23}$$

while the rest of the equations remain unaffected. The reaction constants are the same as the ones used in Section 4.3.1.

The introduced change establishes a negative feedback, i.e., the produced protein blocks the gene that produces it with an intensity proportional to the number of proteins present at a given time.

Results are presented in Figure 4.6. Left and middle panel present results for mRNA and protein that show that in both cases the averages of the stochastic runs (red lines) lie above the deterministic mean (blue lines), albeit the disagreement does not seem important. However, by modifying three parameter values ($H = 0.1$, $K = 60$, $m = 0.01$) the difference of the deterministic and stochastic protein values

differ by a factor of 4 approximately (right panel of same figure).

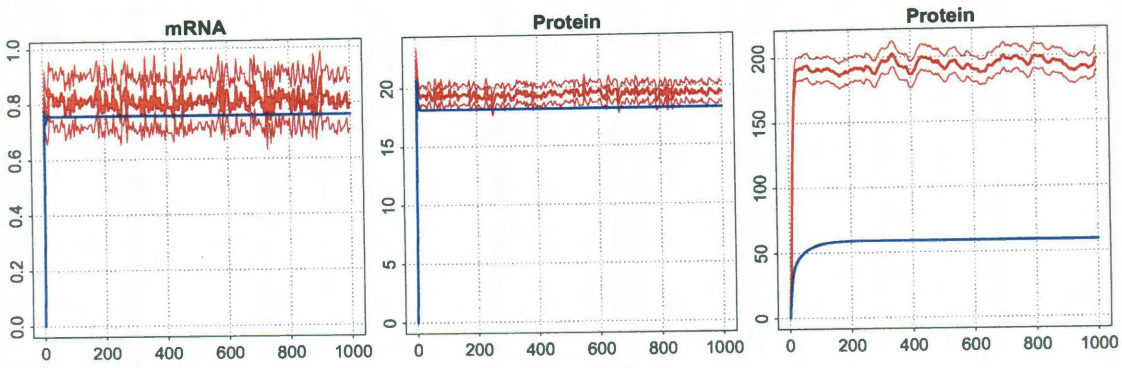


Figure 4.6 : Time evolution of mRNA and protein for gene repressed by its produced protein. Horizontal axis is time units. Vertical axis is number of molecules. Left and middle panel were obtained using $H = 10$, $K = 6$, and $m = 1$, while right panel using $H = 0.1$, $K = 60$, $m = 0.01$.

4.3.4 Gene repressed by dimer of produced protein

We modify the system by:

- Introducing dimerization of the produced protein and dissociation of the dimer.

$$2\ \text{Protein} \xrightarrow{D} \text{Dimer} \tag{4.24}$$

$$\text{Dimer} \xrightarrow{d} 2\ \text{Protein} \tag{4.25}$$

- Replacing the protein by the dimer as the repressor of the gene.

$$\text{Gi} \xrightarrow{c} \text{Ga} + \text{Dimer} \tag{4.26}$$

$$\text{Dimer} + \text{Ga} \xrightarrow{b} \text{Gi} \tag{4.27}$$

Simulations using different values of the introduced parameters D and d do not show a relevant change in structure of the solutions, so the simple arbitrary values $D = 2$ and $d = 1$ are assumed. As already noted, this modified model coincides with the one used to illustrate Petri nets in Chapter 2.

We show results for the dimer in Figure 4.7 using three combinations of values assigned to parameters H and m:

1. the original values ($H = 10$ and $m = 1$), for the case on the left panel;
2. $H = 1$ and $m = 0.1$, for the case on the middle panel,
3. $H = 0.1$ and $m = 0.01$, for the case on the right panel.

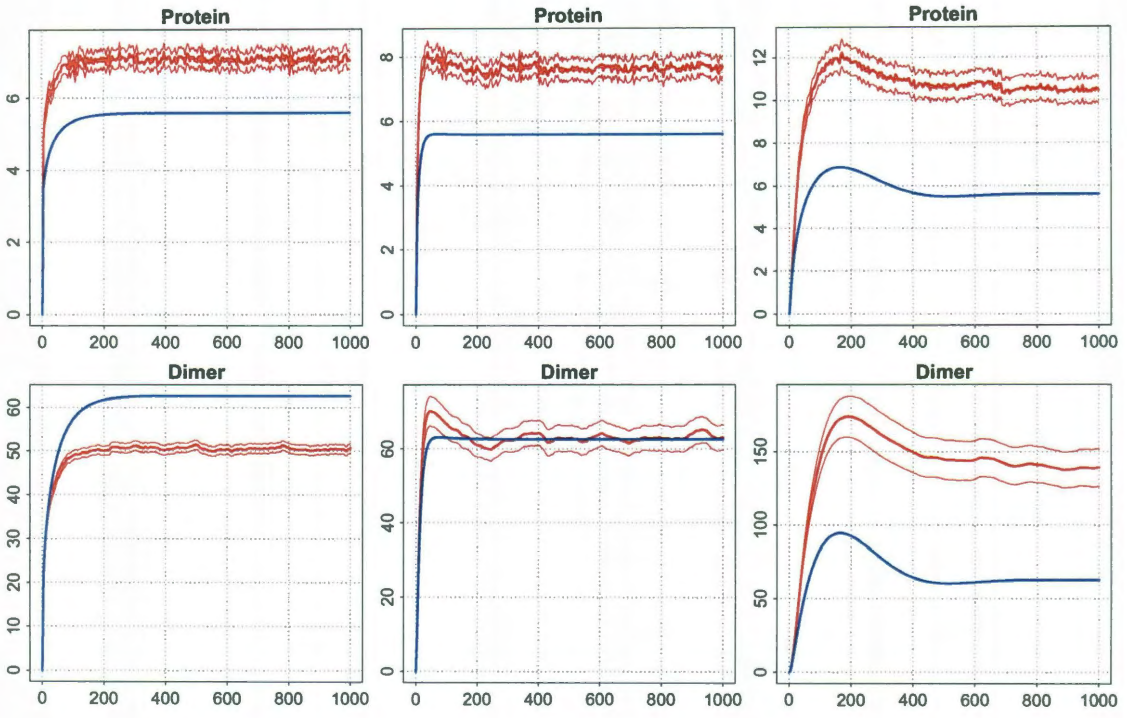


Figure 4.7 : Gene repressed by dimer of produced protein. Time evolution of protein and dimer.

The ratio H/m in all cases is the same ($H/m = 10$). However, the behavior of the system changes: on the left panel the mean of the stochastic runs lies below the deterministic mean, on the middle panel both means agree, and on the right panel it lies above the deterministic mean.

4.4 Bi-stability

We will illustrate the concept of bi-stability using as example published work were bioPN library was successfully used in a large-scale system [58]: the model of cross-talk between regulatory modules of p53 and NF-κB.

Bi-stability emanates when a system has two stable states. Deterministic methods fail to capture this behavior that only stochastic methods (pure of hybrid) can uncover.

We will not describe the model in detail here and refer the reader to the paper for further detail [58]. The model is composed by 46 chemical species and 66 reactions. The system was simulated using the Haseltine and Rawlings hybrid deterministic/stochastic algorithm [27], where gene activation-inactivation, DNA damage and repair, and receptors activation-inactivation were considered slow reactions (15 stochastic reactions), while the rest were considered fast (deterministic). 10,000 runs were performed for three different protocols [58].

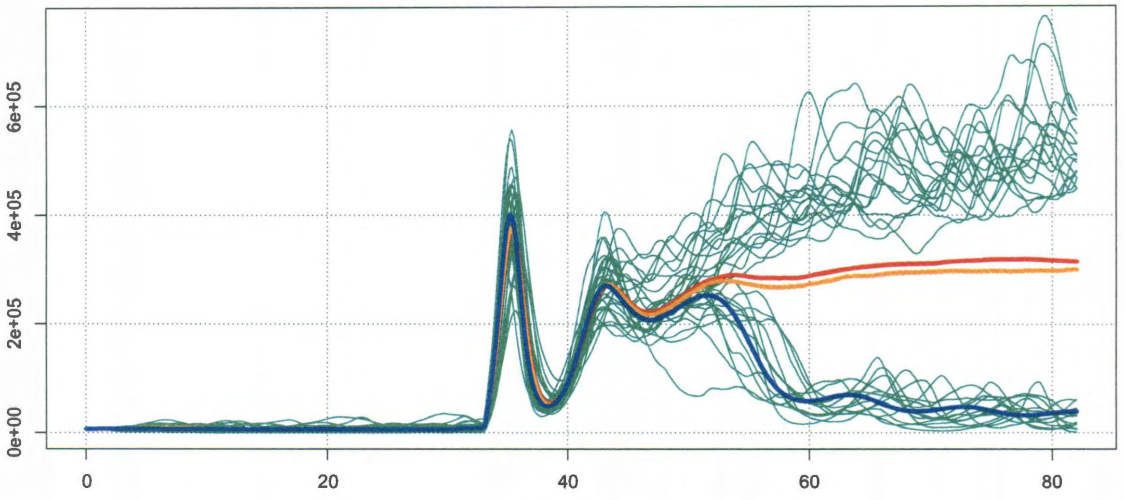


Figure 4.8 : Bi-stability. Comparison of mean values of deterministic and stochastic cases. Horizontal axis is time in hours. Vertical axis is number of units. In all cases it is shown phosphorylated nuclear P53. Green lines: 30 stochastic runs. Red line: average of 1000 stochastic runs. Orange line: average of 1000 hybrid deterministic/stochastic runs. Blue line: deterministic integration

For comparison purposes, we performed two series of 1000 runs, one with the

same hybrid simulation specified above, and the other with a completely stochastic algorithm. The protocol simulated is the following:

1. Cells rest for 30 hours.

2. Cells are stimulated during 4 hours with 10 ng/ml of tumor necrosis factor (TNF), 3 hours before γ irradiation of 2 Gy during 1 hour.

3. Cells are studied for 48 hours of evolution to decide cell fate.

Figure 4.8 shows results for one of the involved proteins, nuclear P53. The system exhibits bi-stability for the chosen parameter values. As expected, the deterministic run (blue line) fails to capture it, and follows only one of the two possible paths.

Figure 4.9 presents another view, and includes on the right column results of the amounts of cells that survive the treatment and of the ones that commit apoptosis. The results for the hybrid and completely stochastic algorithms are very close, which justifies the use of the hybrid algorithm instead of the purely stochastic. The deterministic approach not only fails to capture bi-stability, it also "predicts" that all cells survive for the chosen parameter values, while both the hybrid and stochastic approaches show that the number of cells that become apoptotic (around 60%) is higher that the number of surviving cells (around 40%).

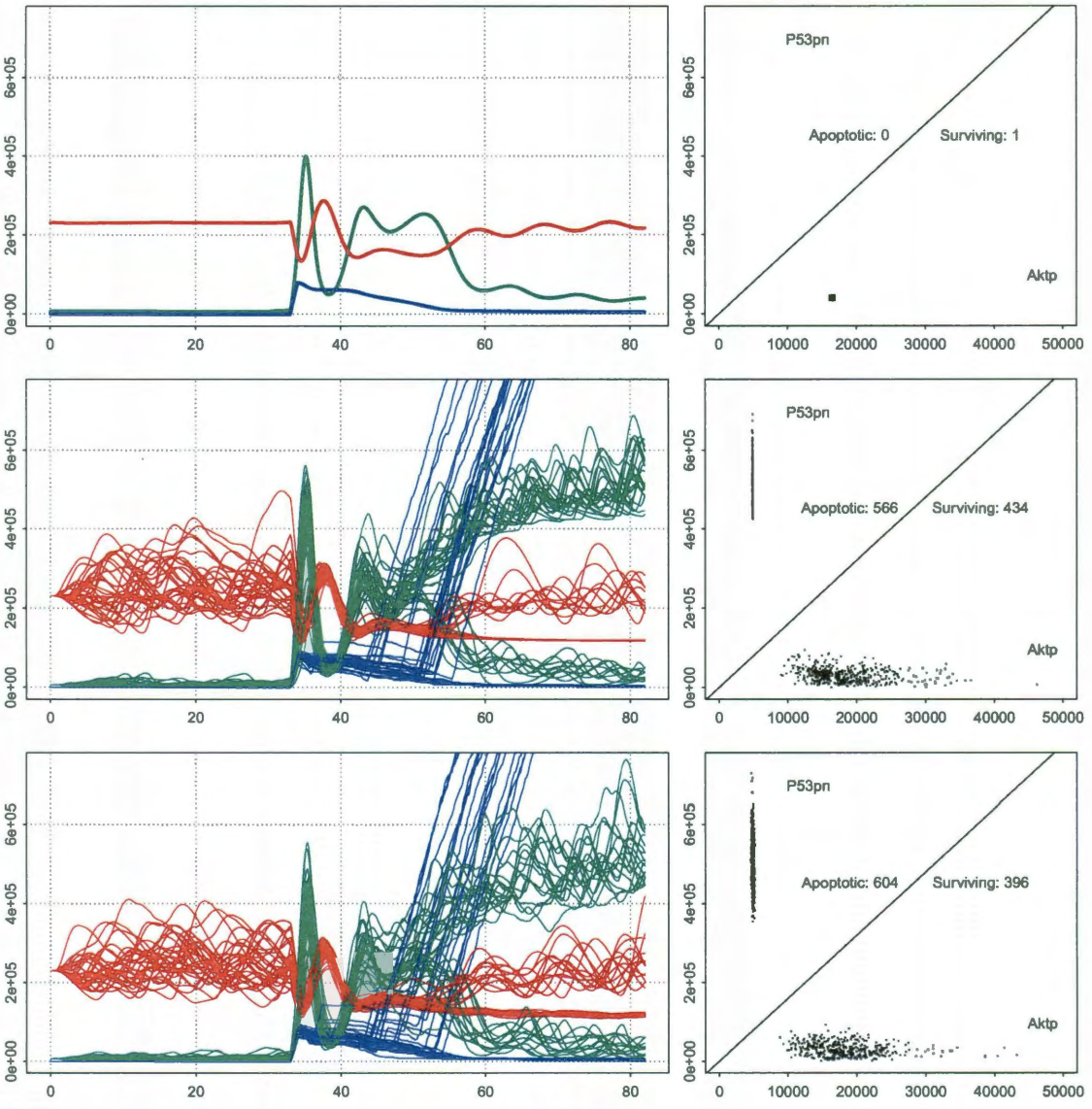


Figure 4.9 : Bi-stability. Comparison of deterministic, hybrid, and stochastic cases (1000 runs). Left panel horizontal axis is time in hours. Left panel vertical axis is number of units. Green lines depict phosphorylated nuclear P53. Red lines depict phosphorylated nuclear Mdm2. Blue lines depict double strand breaks (multiplied by 1000 for comparison purposes). Right panel horizontal and vertical axes are number of units (dots are values at end of simulation). Top row shows results for the deterministic run. Middle row shows runs for the hybrid deterministic/stochastic approach used in the paper. Bottom row shows runs for a completely stochastic approach.

Chapter 5

Transcription factors IRF3 and NFκB acting in a coordinated way under double stranded RNA stimulation

In this chapter we presents current work on a model of cellular sensing of double stranded RNA, which is itself a model of viral infection, by RIG-I protein and TLR3 receptor. It consists of six parts. In part one we introduce the pathway. In part two we describe experiments. In part three we describe experimental results. In part four we introduce the current version of the associated computational model. In part five we describe simulations, protocol used, and simulation results. In part six we compare the outcomes of the deterministic run and of the average of runs of stochastic simulations.

5.1 Introduction

We present current work on a model of cellular sensing of double stranded RNA, which is itself a model of viral infection, by RIG-I and TLR3. Since its inception, the model has undergone five mayor revisions, as experimental results become available. The current version is still a work in progress. The experiments performed verify some of the results found in literature (already reviewed in chapter 3), while others raise the possibility that an alternative explanation may be possible, at least when studying double stranded RNA electroporated into A549 epithelial cells.

The simulations on this study were performed entirely using bioPN, presented in chapter 4. This is study aided by Dr. Allan Brasier group at University of Texas Medical Branch (UTMB), Dr. Tomasz Lipniacki, Institute of Fundamental Technological Research of the Polish Academy of Sciences, and Dr. Marek Kimmel, Rice

University.

Dynamics of innate immunity system under viral attack is still not understood in detail. We provide an introduction to viral infection and innate immune defense with references to literature in Chapter 3. New insights are emerging based both on novel experiments and on system modeling approach. We report a model of coordinated response of two transcription factors, IRF3 and NF-κB, under double stranded RNA stimulation, itself a model for viral RNA.

A brief description of the signaling network follows: double stranded RNA is sensed both by RIG-like family of helicases (RIG-I) and toll-like receptor 3 (TLR3). RIG-I undergoes a conformation change and binds to mitochondrial antiviral signaling adapter (MAVS) and to TNF receptor-associated factor 3 (TRAF3), leading to the nuclear translocation of IFN regulatory factor 3 (IRF3). TLR3 becomes activated, binding to TIR-domain-containing adapter-inducing interferon-β (TRIF) and to TRAF6, triggering phosphorylation and degradation of inhibitor of κB α (IκBα), and freeing an active form of nuclear factor κB (NF-κB) which also translocates into the nucleus. Both IRF3 and NF-κB are independently responsible of the activation of a number of genes involved in innate immune and inflammatory responses. NF-κB also activates the IκBα and tumor necrosis factor, and alpha-induced protein 3 (TNFAIP3, also known as A20) genes that act as inhibitors of the path. Both IRF3 and NF-κB are needed for the activation of the cytokine interferon β.

dsRNA was chosen because it rapidly activates the IRF3 and NF-κB pathways, much more so than, for example, respiratory syncythial virus (RSV), and the timing of dsRNA exposure is more accurate than when working with a virus.

5.2 Experiments

Three kind of experiments were performed at University of Texas Medical Branch (UTMB) by Dr. Allan Brasier's team:

1. Time series of key mRNAs were measured for changes in NFkB and IRF3 signaling pathway. The experiment was performed by Dr. Muping Lu.

2. Time series of key phosphorylated proteins at same time points as above using strong cation exchange (SCX) fractionation experiments, performed by Dr. Yingxin Zhao.

3. Knock-down experiments using small interfering RNA (siRNA) on NF-κB, IRF3, RIG-I, and IKKγ with and without electroporation of dsRNA were performed to study the effect on NF-κB and IRF3 induced genes. The experiment was performed by Dr. Bing Tian.

5.2.1 Knock-down experiments

RelA, IRF3, RIG-I and IKKγ siRNA were transfected into A549 cells using siQUEST reverse transfection method. 4 μg of dsRNA was electroporated into siRNA transfected cells 6 hours before harvesting mRNA. 72 hours after siRNA reverse transfection, the RNA was harvested and reversely transcribed into corresponding cDNA and run for Q-RT-PCR to examine the target gene expression level. The target genes that were examined are:

- RelA, IRF3, RIG-I, and IKKγ, to verify the effect of the knock-downs.
- A20, IκBα, IL8, and IL6, that are considered pure NF-κB dependent genes.
- ISG56, ISG54, CIG5, and ISG60, that are considered pure IRF3 dependent genes.
- IFN and RANTES, that are considered controlled by both IRF3 and NF-κB as their promoter have binding sites for both transcription factors.

5.3 Experimental results

Time series experiments of mRNA and proteins are presented in Figure 5.1, together with simulation results.

5.3.1 Knock-down experimental results

The experimental results, performed in A549 cells, showed that all variants of siRNA could knock-down their corresponding basal gene expression level when compared with the basal gene expression level of control siRNA treated cells (Figure 5.2).

- The basal RelA expression level in RelA siRNA treated cells was 15.8% when compared to control siRNA.
- The basal IRF3 expression level in IRF3 siRNA treated cells was 35% when compared to control siRNA.
- The basal RIG-I expression level in RIG-I siRNA treated cells was 8.2% when compared to control siRNA.
- The basal IKKγ expression level in IKKγ siRNA treated cells was 14.4% when compared to control siRNA.

We observed that dsRNA can sharply increase RIG-I expression level and slightly increase RelA and IRF3 expression levels, while it has no effect on IKKγ expression level.

Results suggest that a cross inhibitory effect between NF-κB and IRF3 pathways could be in place: RelA siRNA treated cells exhibited an elevation in both basal and dsRNA induced levels of IRF3, while IRF3 siRNA treated cells exhibited an elevation in both basal and dsRNA levels of RelA.

However, the effect on the genes induced by both transcription factors is not straightforward:

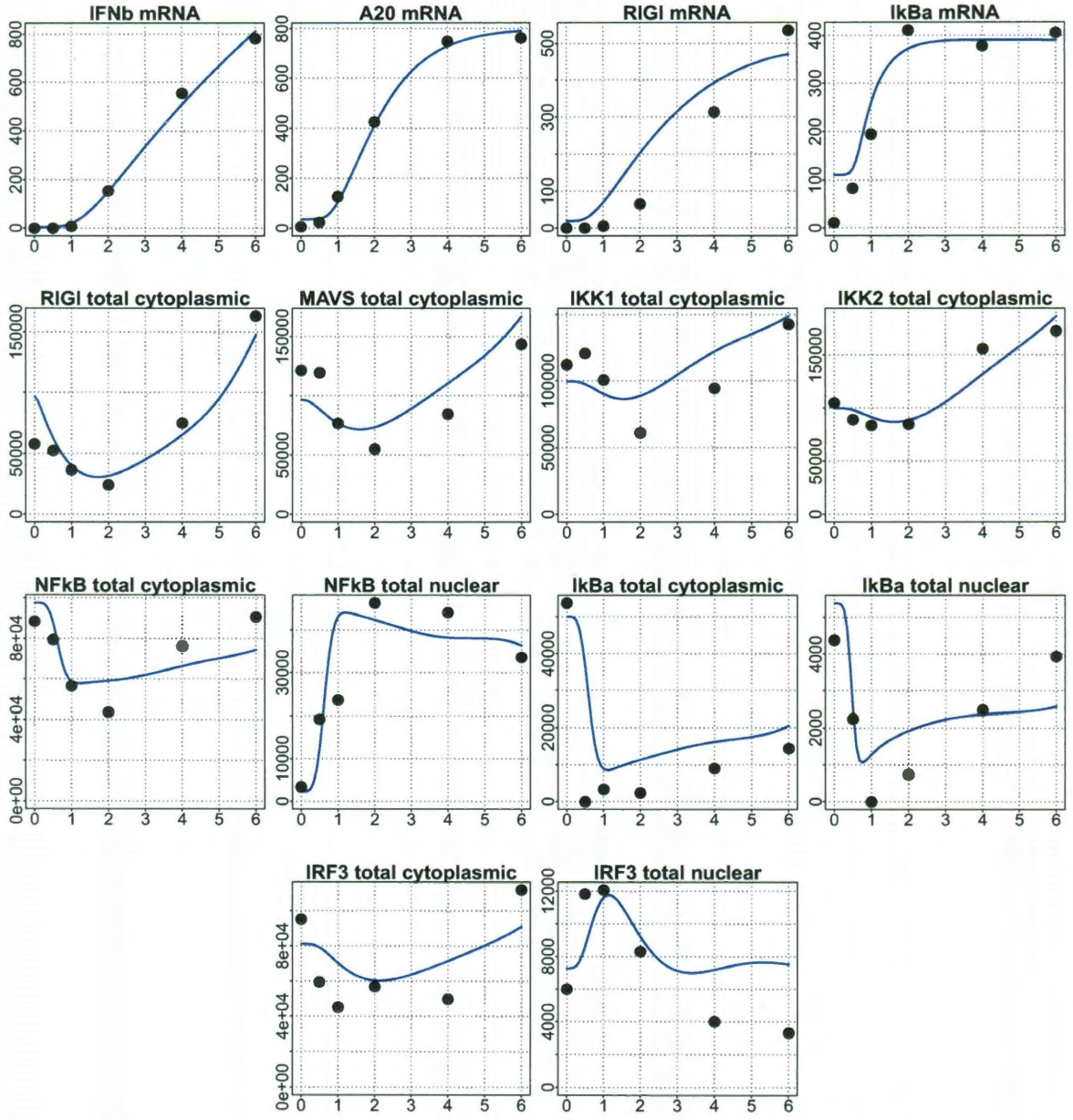


Figure 5.1 : Time evolution of deterministic simulation (blue lines), compared to rescaled experimental results (black dots).

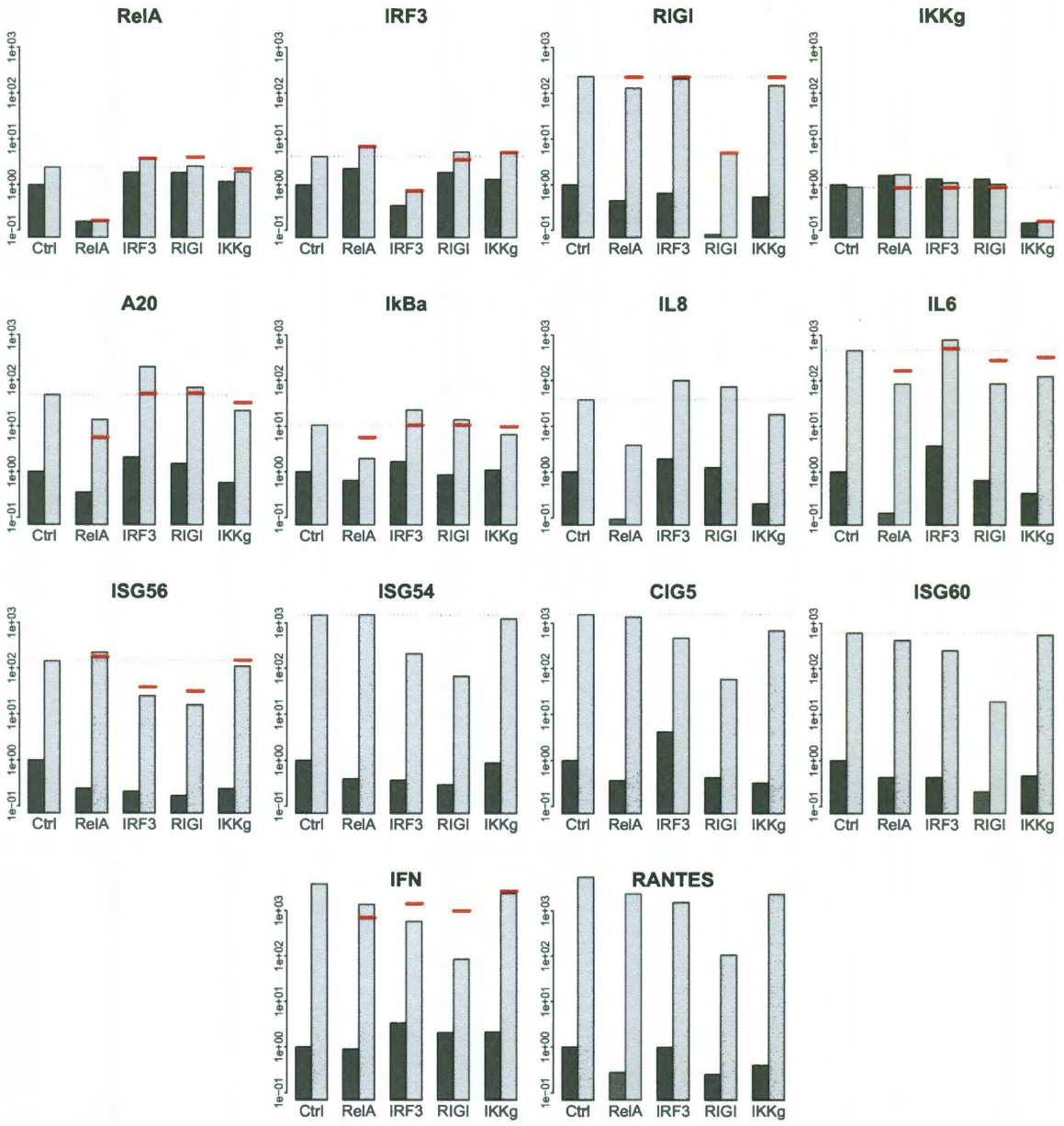


Figure 5.2 : Comparison of knock-down experiments to simulated knock-downs. Simulated values are red lines.

- IRF3 siRNA treated cells exhibited and increase in the expression of RelA controlled genes, such as A20, IκBα, and IL8.

- RelA siRNA treated cells did not exhibited the expected increase in the expression of IRF3 controlled genes: with the exception of ISG56, RelA siRNA did not increase the expression of other IRF3 controlled genes such as ISG54, CIG5, and ISG60.

Dr. Tian notices that, from experience related to previous studies where siRNA was used, in order to have confidence on the experimental results, the level at which a specific siRNA is able to knock down its own gene is critical. Considering that IRF3 siRNA treated cells could knock down IRF-3 gene by 65% of basal IRF3 expression level, Dr. Tian points out we could probably see a stronger effect of IRF3 siRNA treated cells on the level of RelA-dependent gene expression if the previous level (65%) could be increased to 90%. Moreover, the cross inhibitory effect hypothesis could be tested in future experiments, that would require acquisition of RelA, IRF3, RIG-I, and IKKγ deficient cells.

The basal RIG-I expression level in RIG-I siRNA treated cells was 8.2% when compared to control siRNA, which is a satisfactory level. For these experiments, we observe that IRF3 dependent genes are sensitive to the level of RIG-I inhibition, while RelA dependent genes are not. This suggests that dsRNA induced NF-κB pathway activation is probably independent of RIG-I, at least in A549 cells.

5.4 Proposed model

As mentioned before, the current definition of the model is still a draft version. More experiments need to be conducted to validate it, or to suggest further modifications. The model is being constructed from both literature review (Chapter 3), and interpretation of experimental results. A schematic of the model is depicted in Figure 5.3.

The current version of the signaling pathway is composed by 93 chemical species and 173 reactions. We present a simplified schematic where mRNAs and inactive forms of the proteins are omitted for clarity.

The main sensors of dsRNA, as reviewed in 3.4 on page 37, are RIG-I and TLR3. According to literature, both pathways are capable of activating both IRF3 and NF-κB depending on the variant of virus studied, and/or the cell line used. Knock-down experiments presented in the previous section suggest that, in the case of electroporation of dsRNA in A549 cells, RIG-I seems to activate the IRF3 dependent path, but not the NF-κB dependent one.

As a consequence, in our current model we consider two independent paths, one for IRF3 regulated by RIG-I, an one for NF-κB, that we assume regulated by TLR3, as literature supports. However, further experiments with TLR3 knocked-down should be performed to assess the validity of this assumption.

Certain complexes have been named to simplify notation:

- The complex TBK1-IKKϵ is termed IKK1 and simplified as a single kinase that has a gene and mRNA associated.
- The same treatment is done for the complex IKKα-IKKβ-IKKγ, termed IKK2.
- TLR3 dependent transcription factor (TLR3dTF) is an hypothetical transcription factor activated exclusively by TLR3-TRIF-TRAF6 pahtway. Our main candidate is AP-1.

Full coupled chemical reactions are presented on section C.2.

5.5 Simulations

The model was simulated using bioPN. Knock down simulations are performed by increasing the corresponding mRNA degradation rate by a factor that achieves the same level of down-regulation as shown in the experimental results.

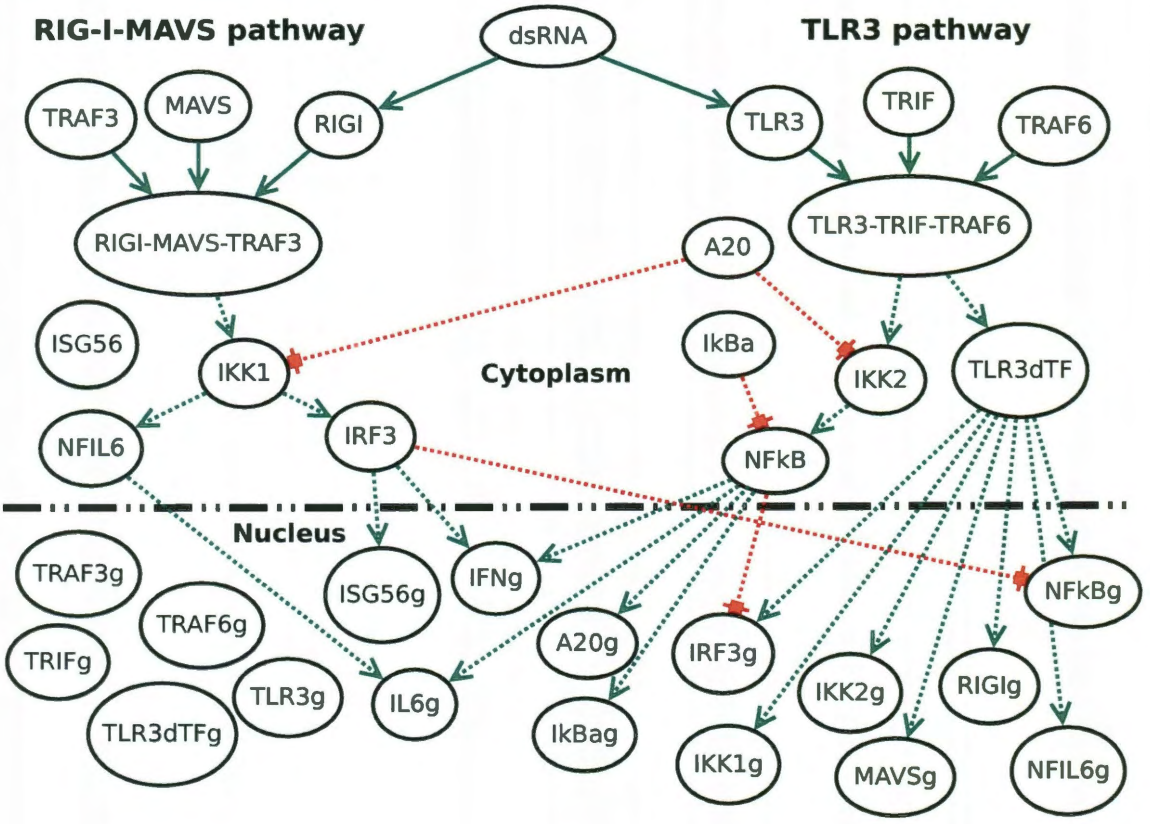


Figure 5.3 : Simplified schematics of IRF3-NFκB model. Only proteins (in the cytoplasm) and genes (in the nucleus) are shown. Green lines imply activation when destination is in cytoplasm and up-regulation when in nucleus. Solid green lines on top mean dsRNA activates while binding to destination. Red dotted lines imply inhibition. Transcripts and inactive forms of the proteins, as well as nuclear translocated proteins, are omitted to avoid cluttering.

5.5.1 Protocol

Initial conditions are set to zero number of molecules for all species, with the exception of the number of free (not bound to transcription factor) genes, which is set to 2 copies. The simulation is started at time $t = -500$ hours (-100 hours would have been more than enough) to assure the system reaches steady state values. At time $t = 0$ hours, electroporation is simulated by inserting 5×10^5 molecules of dsRNA, which corresponds to the experimental values of 4 μg. The system is allowed to evolve for 6 hours.

5.6 Simulation results

5.6.1 Simulations of full model experiments

Simulation results of full model experiments are presented in Figure 5.1. The upper panel shows the fit of the model to mRNA time series of experimental results after rescaling the experimental data using least squares. The rescaling is needed because the absolute quantification of mRNA as well as of proteins is, in many situations, irreconcilable with physiological levels (as an example, 10^5 molecules of RIG-I mRNA per cell).

The rest of the panels show the fit of the model to protein time series of experimental results, again after rescaling of experimental results by least squares.

5.6.2 Simulations of knock-down experiments

Figure 5.2 shows, in each panel, the effect of all knock-down on a specific gene expression. The bars in these figures are the results of the experimental results, and the red segments are the corresponding simulation values at six hours under dsRNA stimulation. The vertical axis is in log-scale with a fixed maximum to comparison purposes across panels. A different presentation can be found in the Appendix C (page 180).

5.6.3 Comparison of simulations with experiments

The fit of the current model to the experimental data shows good agreement in most cases, while in some it does not agree. There is a trade off when modifying coefficients, and we are interested in achieving a global agreement, even if it may not work for a particular case. More experiments are needed and planned. The complete set of experiments will provide a better understanding of the dynamics involved, and may suggest modifications to the actual structure of the model.

5.7 Comparison between deterministic and stochastic systems

The experiments performed involve the introduction of a high amount of dsRNA (in the order of 5×10^5), which provokes a strong reaction on the signaling pathways. As such, we performed this study deterministically by numerically solving the corresponding set of ordinary differential equations. In this section we will compare the deterministic results with single stochastic runs and the averages of 1000 stochastic runs, to see the level of agreement between both approaches.

To shorten the time needed to run the stochastic simulations, the initial run for this series of simulations is reduced to 100 hours, as our intention is to compare both deterministic and stochastic methods, and not to assure that all species are at steady state values.

Results are presented in Figure 5.4. As in previous chapters, blue lines represent deterministic results, orange lines represent single stochastic runs, and red lines represent averages of 1000 stochastic runs. We have included in these plots the standard deviations of 1000 stochastic runs, depicted in red dotted lines. We show only protein results. On the left it is depicted the evolution before dsRNA electroporation, starting from initial conditions set to two copies of inactive gene for each species, and zero for the rest of the components. On the right we show 30 hours of evolution since dsRNA electroporation, modeled by introducing at time zero 5×10^5 molecules of dsRNA. This amount of dsRNA is depleted in about 8 hours, after which the system evolves to a similar behavior as before treatment.

TRIF protein is depicted on the top panel. In our model TRIF has an unregulated gene associated. As such, the reactions that lead to its creation and degradation are of first order, so deterministic and stochastic approaches should agree, (as we have established in Section 2.3.4 on page 17). Numerical simulations verify this and the agreement between both is remarkable (the deterministic blue line covers the

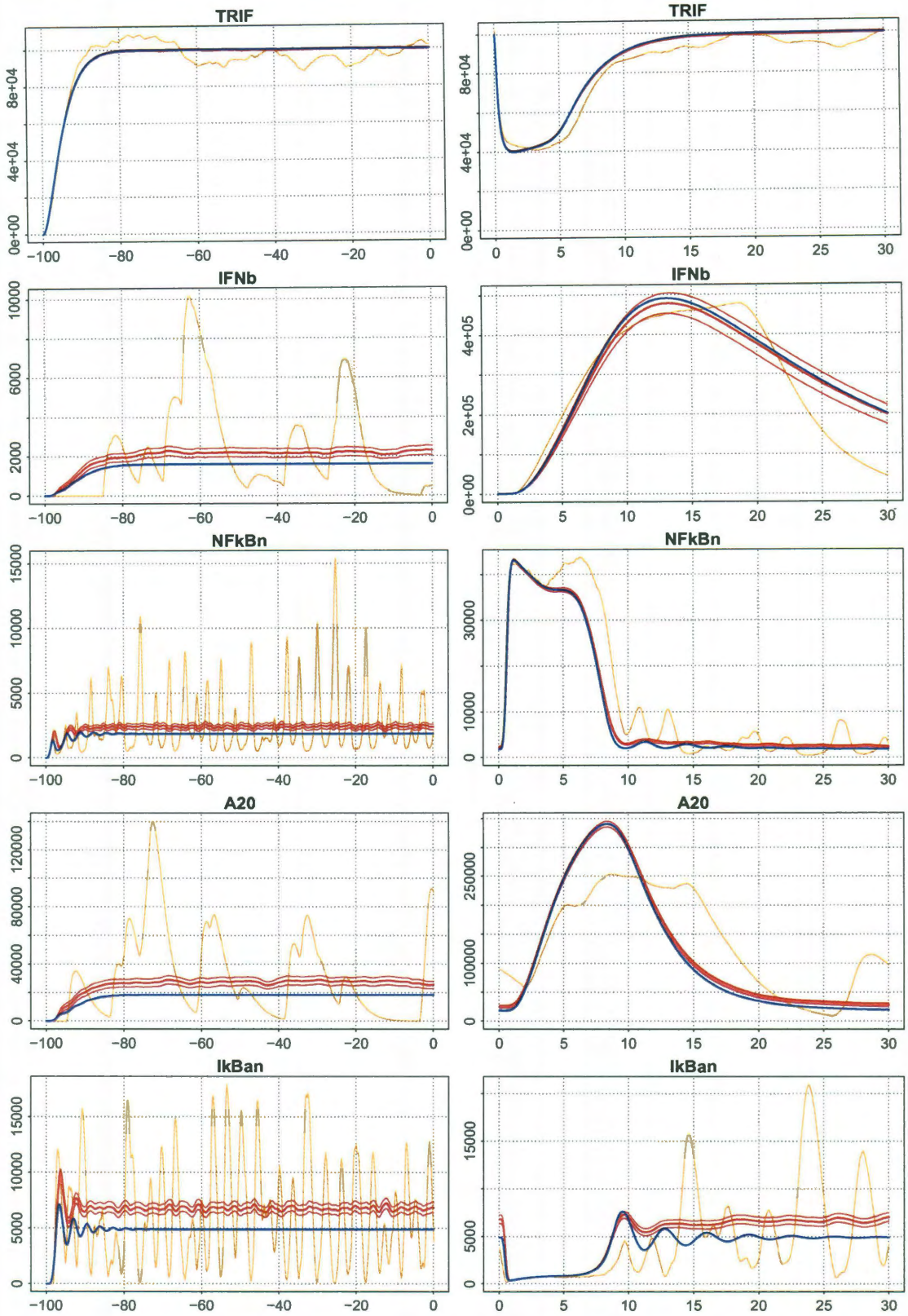


Figure 5.4 : Comparison between deterministic and stochastic systems for IRF3/NF-κB signaling pathway. Left panel shows evolution of the system for 100 hours before dsRNA treatment, while right panel shows the evolution of 30 hours from dsRNA treatment administration.

stochastic red line at all times).

In the structure of the model, the majority of the proteins have associated genes that are regulated by transcription factors (IRF3 and TLR3dTF) that only get activated as a consequence of dsRNA electroporation. As such, these genes are not active in the initial 100 hours and produce mRNA only at basal level. Therefore, their corresponding proteins act as produced by unregulated genes. We omit showing these cases, as they look a similar concordance as the observed for TRIF.

In the model, NF-κB is a protein tightly regulated by IκBα and A20. Before treatment A20 does not play a regulating role, but IκBα does. As a consequence, proteins regulated by NF-κB have interesting dynamics, that can be observed on the left column on second to fourth rows of the figure. In these shown cases, the disagreement between deterministic and stochastic results is evident in levels, and in certain cases, in dynamics.

The strong stimulating effect produced by dsRNA electroporation seems to overcome any other effect, and deterministic and stochastic results closely agree during the first hours after treatment for all species, as it can be observed on the right column. At later times each species evolves to similar behavior as it had before treatment.

Part II

Reaction-diffusion systems, sbioPN and applications to early carcinogenesis and viral infection

Chapter 6

Reaction-diffusion modeling background

In this chapter we introduce reaction-diffusion systems. It consists of two parts. In part one we discuss the importance of considering spatial effects, such as diffusion, as well as incorporating stochasticity in models where a deterministic approach is not a viable solution. In part two we summarize the mathematical background relevant to modeling of reaction-diffusion systems, and deterministic and stochastic numerical approximations.

6.1 Introduction to reaction-diffusion systems

The methods used so far in the studied systems assume the reactants are perfectly mixed and new species, created by reactions, are also instantaneously mixed in a perfectly homogeneous medium where position is irrelevant. This simplification eases the mathematical treatment of the system, and this approach has proved adequate to solve numerous problems.

However, sometimes spatial patterns are present and cannot be neglected without incurring in an over-simplification far apart from reality. In these cases reaction-diffusion methods need to be used.

There are two approaches to the mathematical modeling of chemical reaction-diffusion systems: deterministic models based on systems of ordinary and partial differential equations; and stochastic simulations [15].

Deterministic methods are commonly used when high number of molecules are present and the interest is in studying effects averaged over the cell population. We have already discussed the implications of using deterministic methods on stochastic non-spatial systems in Part I. In reaction-diffusion systems where low copy numbers

of molecules or cells are present, deterministic models are inaccurate or even it is not possible to apply them [15]. In these cases stochastic modeling provides a more detailed understanding of the reaction-diffusion processes.

Reaction-diffusion approaches may me suitable to study biological systems. We are interested in the effects of reactions and diffusion within- and among-cells.

The first case is applicable to situations where a cell cannot be simplistically conceived as a perfectly mixed tank. One example is the interaction between a cytoplasmic protein and a receptor that is located on the boundary of the cell. Only proteins that are close to the boundary are in position to bind to the receptor. The heterogeneity present in this system cannot be neglected. Moreover, if only few copies of the receptor are present, treating the system deterministically would be incorrect.

The second case is applicable when different cells signal each other through cytokines, or to model the effect on cells of viral infection. Again, stochasticity may play a major role. We aim to explore the importance of stochastic effects in these type of models.

6.2 Background on modeling reaction-diffusion systems

6.2.1 Diffusion

Diffusion is the effect of random movement of molecules in a medium due to thermal energy [5]. The governing law is Brownian motion (see Def 6.1). There is a unique *diffusion constant* D that characterizes diffusion, that depends on the size of the molecule, viscosity and absolute temperature of the solution. The value of D for a typical protein molecule is $D = 10^{-4}\text{mm}^2\text{sec}^{-1}$ [15].

In Cartesian coordinates, let $M(x, y, z, t)$ be the number of molecules (or concentration) of a given species at point (x, y, z) at time t. Given initial conditions $M(x, y, z, 0)$, a diffusive system will behave according to the partial differential equation (PDE) [61], also known as the *heat equation*:

$$\frac{\partial M}{\partial t} = D\nabla^2 M = D\left(\frac{\partial^2 M}{\partial x^2} + \frac{\partial^2 M}{\partial y^2} + \frac{\partial^2 M}{\partial z^2}\right) \tag{6.1}$$

where ∇^2 is the Laplace operator, defined in 6.3.2 for different systems of coordinates.

6.2.2 Deterministic approximation of diffusion

6.2.2.1 One spatial dimension

We will first solve the deterministic problem. Limiting the analysis to one dimension, the previous expression simplifies into

$$\frac{\partial M}{\partial t} = D\frac{\partial^2 M}{\partial x^2} \tag{6.2}$$

To numerically solve this PDE, we can break the continuous space into discrete spatial points: $0, \Delta x, 2\Delta x, \cdots, i\Delta x, \cdots$, defining $x_i = i\Delta x$. Also, we discretize the time in Δt intervals. Calling $M_j(x_i)$ the approximation to $M(i\Delta x, j\Delta t)$, we can numerically integrate this system using the iterative forward Euler method:

$$M_{n+1}(x_i) = M_n(x_i) + \Delta t\frac{\partial M_n(x_i)}{\partial t} = M_n(x_i) + \Delta t D\frac{\partial^2 M_n(x_i)}{\partial x^2} \tag{6.3}$$

Estimating $\frac{\partial^2 M_n(x_i)}{\partial x^2}$ using a centered difference approximation [61], we obtain:

$$M_{n+1}(x_i) = M_n(x_i) + \Delta t D\frac{M_n(x_{i+1}) + M_n(x_{i-1}) - 2M_n(x_i)}{\Delta x^2} \tag{6.4}$$

Suitable initial conditions $M_0(x_i)$ need to be defined for all discretized spatial points, as well as suitable boundary conditions.

Boundary conditions can be of two types: *Dirichlet boundary conditions* specify the values at the boundaries for each time t, and can be a constants or variable functions; *Neumann boundary conditions* are imposed on the derivatives of the values

at the boundaries. For this last case, the centered difference approximation cannot be used, as the values at the extremes are undefined. For example, to update x_1, the following approximation can be used for $\frac{\partial^2 M_n(x_1)}{\partial x^2}$:

$$\frac{\partial^2 M_n(x_1)}{\partial x^2} \approx \frac{M_n(x_2) - M_n(x_1) - \Delta x \frac{\partial M_n(x_0)}{\partial x}}{(3/2)\Delta x^2} \tag{6.5}$$

where $\frac{\partial M_n(x_0)}{\partial x}$ is the Neumann boundary condition at the left extreme.

Rearranging (6.4) as

$$\frac{M_{n+1}(x_i) - M_n(x_i)}{\Delta t} = D\frac{M_n(x_{i+1}) + M_n(x_{i-1}) - 2M_n(x_i)}{\Delta x^2} \tag{6.6}$$

we can interpret each side as an approximation to a derivative:

$$\frac{\partial M}{\partial t} + O(\Delta t) = \frac{\partial^2 M}{\partial x^2} + O(\Delta x^2) \tag{6.7}$$

which suggests that, in general, we want $\Delta t \sim \Delta x^2$ so that both sources of error have similar magnitudes.

6.2.2.2 Three spatial dimensions

Discretizing equation (6.1) we obtain

$$\begin{aligned} &\frac{M_{n+1}(x_i, y_j, z_k) - M_n(x_i, y_j, z_k)}{\Delta t} = \\ &D\frac{M_n(x_{i+1}, y_j, z_k) + M_n(x_{i-1}, y_j, z_k) - 2M_n(x_i, y_j, z_k)}{\Delta x^2} \\ &+D\frac{M_n(x_i, y_{j+1}, z_k) + M_n(x_i, y_{j-1}, z_k) - 2M_n(x_i, y_j, z_k)}{\Delta y^2} \\ &+D\frac{M_n(x_i, y_j, z_{k+1}) + M_n(x_i, y_j, z_{k-1}) - 2M_n(x_i, y_j, z_k)}{\Delta z^2} \end{aligned} \tag{6.8}$$

for which initial conditions and suitable boundary conditions need to be defined. Also we want $\Delta t \sim \Delta x^2 \sim \Delta y^2 \sim \Delta z^2$.

6.2.3 Reaction-diffusion equations

The general form of a reaction-diffusion equation is:

$$\frac{\partial M}{\partial t} = D\nabla^2 M + f(M) \tag{6.9}$$

where M is a vector of places (species), D is a diagonal matrix with the diffusion rate of each species ($D\nabla^2 M$ is a vector called the diffusion term), $f(M)$ is a vector where the elements are the standard system of mass-action differential equations, and it is called the reaction term.

6.2.4 Stochastic approximation of diffusion

6.2.4.1 Discretization of stochastic differential equation approach

Let $[X(t), Y(t), Z(t)] \in \mathbb{R}^3$ be the position of a diffusing molecule at time t. Choosing a time step Δt, the following discretized versions of the stochastic differential equations (SDEs) (defined in Section 6.3.6) can be used to track the evolution of diffusing single molecules [15]:

$$X(t + \Delta t) = X(t) + \sqrt{2D\,\Delta t}\,\xi_x \tag{6.10}$$

$$Y(t + \Delta t) = Y(t) + \sqrt{2D\,\Delta t}\,\xi_y \tag{6.11}$$

$$Z(t + \Delta t) = Z(t) + \sqrt{2D\,\Delta t}\,\xi_z \tag{6.12}$$

where ξ_x, ξ_y, and ξ_z are $\mathcal{N}(0, 1)$.

Suitable boundary conditions need to be defined. These equations are related to the chemical Langevin equations, introduced in Section 2.5.2.

6.2.4.2 Compartment-based approach

We divide the computational domain $[0, L]$ into K compartments of length $h = L/K$. As long as, regardless of dimension [15, 33],

$$d = \frac{D}{h^2} \tag{6.13}$$

the following chain of first order reactions will be a correct approximation for simulating diffusion with a stochastic simulation algorithm (SSA):

$$A_1 \underset{\mathrm{d}}{\overset{\mathrm{d}}{\rightleftharpoons}} A_2 \underset{\mathrm{d}}{\overset{\mathrm{d}}{\rightleftharpoons}} A_3 \underset{\mathrm{d}}{\overset{\mathrm{d}}{\rightleftharpoons}} \cdots \underset{\mathrm{d}}{\overset{\mathrm{d}}{\rightleftharpoons}} A_K \tag{6.14}$$

6.3 Mathematical definitions and numerical methods.

6.3.1 Del, gradient and divergence operators

Let us limit for now to n-dimensional Euclidean space to introduce some preliminary concepts.

Del (∇) is a vector differential operator, defined as:

$$\nabla = \left(\frac{\partial}{\partial x_1}, \cdots, \frac{\partial}{\partial x_n}\right) = \frac{\partial}{\partial x_1}\mathbf{e_1} + \cdots + \frac{\partial}{\partial x_n}\mathbf{e_n} \tag{6.15}$$

where $\{\mathbf{e_i} : i = 1, \cdots, n\}$ is the standard basis in this space.

Applying del to a *scalar field* f (a function that for each point of the space returns a scalar) is defined as the *gradient* of the scalar field:

$$\nabla f = \left(\frac{\partial f}{\partial x_1}, \cdots, \frac{\partial f}{\partial x_n}\right) = \frac{\partial f}{\partial x_1}\mathbf{e_1} + \cdots + \frac{\partial f}{\partial x_n}\mathbf{e_n} \tag{6.16}$$

The gradient converts a scalar field into a vector.

Given a vector $\mathbf{V} = (v_1, \cdots, v_n) = v_1\mathbf{e_1} + \cdots + v_n\mathbf{e_n}$, the *divergence* ($\nabla \cdot$) is an

operator that first applies the components of del to each of the components of the vector, and then sums the components (reminiscent to a dot product), giving a scalar as a result. The representation $\nabla\cdot$ is considered an abuse of notation. In mathematical notation:

$$\nabla \cdot \mathbf{V} = \left(\frac{\partial}{\partial x_1}\mathbf{e_1} + \cdots + \frac{\partial}{\partial x_n}\mathbf{e_n}\right) \cdot (v_1\mathbf{e_1} + \cdots + v_n\mathbf{e_n}) = \frac{\partial v_1}{\partial x_1} + \cdots + \frac{\partial v_n}{\partial x_n} \quad (6.17)$$

6.3.2 Laplace operator

Given n-dimensional Euclidean space, the *Laplace operator* or *Laplacian* is a second order differential operator, defined as the *divergence* ($\nabla\cdot$) of the *gradient* of a function (∇f).

If f is a twice differentiable real-valued function, the *Laplacian* of f (Δf) is defined as:

$$\Delta f = \nabla^2 f = \nabla \cdot \nabla f \quad (6.18)$$

In n-dimensional *Cartesian* coordinates:

$$\Delta f = \nabla \cdot \nabla f = \nabla \cdot \left(\frac{\partial f}{\partial x_1}, \cdots, \frac{\partial f}{\partial x_n}\right) = \sum_{i=1}^{n} \frac{\partial^2 f}{\partial x_i^2} \quad (6.19)$$

Considering $n = 3$ Cartesian coordinates:

$$\Delta f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} + \frac{\partial^2 f}{\partial z^2} \quad (6.20)$$

In *cylindrical* coordinates:

$$\Delta f = \frac{1}{\rho}\frac{\partial}{\partial \rho}\left(\rho \frac{\partial f}{\partial \rho}\right) + \frac{1}{\rho^2}\frac{\partial^2 f}{\partial \theta^2} + \frac{\partial^2 f}{\partial z^2} \quad (6.21)$$

In *spherical* coordinates:

$$\Delta f = \frac{1}{r^2}\frac{\partial}{\partial r}\left(r^2\frac{\partial f}{\partial r}\right) + \frac{1}{r^2\sin\theta}\frac{\partial}{\partial\theta}\left(\sin\theta\frac{\partial f}{\partial\theta}\right) + \frac{1}{r^2\sin^2\theta}\frac{\partial^2 f}{\partial\varphi^2} \tag{6.22}$$

where θ is the *polar* angle and φ the *azimuthal* angle.

If $n = 2$, in Cartesian coordinates:

$$\Delta f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} \tag{6.23}$$

whereas in *radial* coordinates:

$$\Delta f = \frac{1}{\rho}\frac{\partial}{\partial\rho}\left(\rho\frac{\partial f}{\partial\rho}\right) + \frac{1}{\rho^2}\frac{\partial^2 f}{\partial\theta^2} \tag{6.24}$$

6.3.3 Brownian motion

Definition 6.1 Brownian motion (or a standard Wiener process) [29] is a random process

$$W(t) \quad (t \in [0,T]) \tag{6.25}$$

for which:

1. $W(0) = 0$ a.s.
2. if $0 \le s < t \le T$ then $W(t) - W(s) \sim \sqrt{t-s}\,\mathcal{N}(0,1)$
3. if $0 \le s < t \le u < v \le T$ then $W(t) - W(s)$ and $W(v) - W(u)$ are independent.

6.3.4 Discretized Brownian motion

It is usually considered for computational purposes. We set $\delta t = T/N$ for some positive integer N and let $W_j = W(t_j = j\,\delta t)$. Then $W_0 = 0$ a.s., and

$$W_j = W_{j-1} + dW_j \quad (j = 1, 2, \ldots, N) \tag{6.26}$$

where dW_j are iid distributed as $\sqrt{\delta t}\mathcal{N}(0,1)$.

6.3.5 Stochastic integral

The stochastic integral of the function $h(t)$ with respect to a Brownian motion $W(t)$ can be defined as an *Itô* integral

$$\int_0^T h(t)\,dW(t) = \lim_{N\to\infty} \sum_{j=0}^{N-1} h(t_j)(W(t_{j+1}) - W(t_j)) \tag{6.27}$$

or a *Stratonovich* integral

$$\int_0^T f(W(t)) \circ dW(t) = \lim_{N\to\infty} \sum_{j=0}^{N-1} h\left(\frac{t_j + t_{j+1}}{2}\right)(W(t_{j+1}) - W(t_j)) \tag{6.28}$$

Note that the two stochastic sums give different answers, no matter how small the time step.

A Stratonovich integral can be expressed as an Itô integral as

$$\int_0^T f(W(t)) \circ dW(t) = \frac{1}{2}\int_0^T f'(W(t))\,dt + \int_0^T f(W(t))\,dW(t) \tag{6.29}$$

6.3.6 Stochastic differential equations (SDE)

A scalar, autonomous stochastic differential equation (SDE) can be written

$$dX(t) = f(X(t))dt + g(X(t))dW(t), \quad X(0) = X_0, \quad 0 \le t \le T \tag{6.30}$$

The integral form is

$$X(t) = X_0 + \int_0^t f(X(t))ds + \int_0^t g(X(s))dW(s), \quad 0 \le t \le T \tag{6.31}$$

where f and g are scalar functions, X_0 a random variable, and $X(t)$ a random variable for each t.

6.3.7 Euler-Maruyama (EM) method

To numerically solve the previous SDE over $[0, T]$, we discretize the interval, setting $\Delta t = T/L$ for a positive integer L, and $\tau_j = j\,\Delta t$. Denoting $X(\tau_j) = X_j$, the EM method is:

$$X_j = X_{j-1} + f(X_{j-1})\Delta t + g(X_{j-1})(W(\tau_j) - W(\tau_{j-1})), \quad j = 1, 2, \ldots, L \qquad (6.32)$$

$W(\tau_j) - W(\tau_{j-1})$ will be obtained from the discretized Brownian motion. For convenience $\Delta t = R\,\delta t$ for an integer $R \geq 1$.

Chapter 7

sbioPN, the reaction-diffusion extension to bioPN library

In this chapter we present sbioPN (spatial bioPN), a software library developed to solve reaction-diffusion systems in deterministic, stochastic, and hybrid deterministic/stochastic manner. It consists of six parts. In the first part we discuss the limitations of bioPN to simulate reaction-diffusion systems. In the second part we present sbioPN and its characteristics. In the third part we explain the structures needed to define a reaction-diffusion model in sbioPN. In the fourth part we compare diffusion in deterministic and stochastic settings. In the fifth part we make some considerations about compartment size in cubic grids. Finally, in the sixth part we shows two applications from literature simulated with sbioPN.

7.1 Introduction

bioPN, the library presented in chapter 4, could in principle be used to simulate reaction-diffusion systems, partitioning the container in compartments and creating instances for each chemical species in each compartment, as reviewed in Chapter 6 (Subsection 6.2.4.2 on page 80).

For example, we may call the instances of chemical species A A_1, A_2, A_3 (species A in compartment 1, 2 and 3). Proceeding this way, we should repeat each specific reaction as many times as different compartments are considered, modifying the instances of the chemical species that react (using A_1 for the first reaction, A_2 for the second, and so on). Following the principles of compartment-based approach, diffusion can be modeled as first order reactions (A_1 transforming into A_2 and vice versa, and so on), with the reaction rate constant as defined in Equation 6.13.

However, this approach could only be considered successful in small systems, and would be error prone. The use of an algorithmical approach to create the instances of the reaction could be a solution. However, in systems with a large number of compartment, sparse matrices *Pre* and *Post* could potentially exhaust the available memory during the definition of the model.

7.2 Characteristics of sbioPN

sbioPN, for spatial bioPN, is an extension to bioPN that can be used to simulate reaction-diffusion systems. By design it is not tied to any specific spatial configuration, even if it has been used up to date only for one- and two-dimensional Cartesian systems, where the compartment, called voxel in sbioPN, is considered, respectively, a segment or a square. Three dimensional Cartesian system is a natural extension for which sbioPN is already prepared. Theoretically, sbioPN could be used for cylindrical or spherical coordinate systems, and the container been considered an hexagon, or a ring, or a hollow sphere, according to the case. This is possible because there are two kind of reactions considered:

- reactions that occur inside a specific container and depend on species only belonging to that specific container.
- reactions that involve the migration of a species from one container to another.

Only the second type of reaction requires awareness of the spatial configuration. A system that only has transitions of the first type can be conceived as several independent simulations performed at once, with the same type of outcome that would be obtained by performing a sequential series of runs of a non-spatial system.

It is then a correct definition of the migration rules which suffices to unambiguously define the model. However, more study is needed to correctly define the reaction rates for models in coordinate systems other that Cartesian, so we will limit our attention

in this thesis to one- or two-dimensional. We plan on improving sbioPN so it can simulate reaction-diffusion processes on unstructured meshes [13].

sbioPN is not limited by design to diffusive systems. It could potentially be used to solve, for example, reaction-convective systems, or any other conceivable law of migration. A data structure with rules detailing how species migrate is used. This gives the modeler the possibility to design the desired scheme. sbioPN allows to implement different sets of migration rules for different species.

We have only used sbioPN so far to solve reaction-diffusion systems, and we provide in the Appendix code to create rules for one and two dimensional Cartesian grids with Neumann and periodic boundary conditions.

As happens with bioPN, sbioPN allows to define the model once, and then decide which function to call depending if a deterministic, stochastic, or hybrid deterministic/stochastic simulation is desired. Following bioPN, functions that perform simulations are programmed in C and optimized for performance.

7.3 Model definition in sbioPN

7.3.1 Structures needed to define the model

To define the model, an R list with the following named elements is constructed.

1. **Pre** and **Post** matrices (required). These matrices are defined as in bioPN. Their interpretation changes in the fact that now the places and transitions are general. The library generates instances of each place (column) for each container. Each transition (row) is not required to occur in every container; it can occur in any subset of containers.

2. **h** list (required). Same as in bioPN.

3. **M** matrix (required), used to set the initial conditions. It has as many rows as containers and columns as places. If only one container is defined, it can be a

vector.

4. **Voxels** scalar (required), indicating the total number of containers in the model.

5. **VoxelSubset** list (required). Each element of the list is a vector, whose elements are the indexes corresponding to the containers included on a particular subset.

6. **TransitionInVoxelSubset** vector (required). It is used to assign transitions to subsets of containers. It has as many elements as transitions, and each element is an index to the **VoxelSubset** list defined above.

7. **JumpingPattern** list (optional). Each element of the list is a matrix with 3 columns and as many rows as jumps this particular jumping pattern will have. The first column indicates the index of the source container of the jump, while the second the index of the destination one. The third column indicates the stoichiometry of the post matrix. See 7.3.2 for further explanation.

8. **JumpingPlace** list (optional). Used to indicate which places migrate. Each element is a list with four elements:

   - Place
   - jumping rate
   - index to the **JumpingPattern** list defined above.
   - 1 if the reaction is slow or 0 if it is fast (for Haseltine and Rawlings function only).

   The same place may appear more than once, with different jumping rate and jumping pattern.

7.3.2 JumpingPattern structure

A successful definition of the jumping patterns is required, as those patterns are the ones that confer the spatial characteristic to the model, as well as the law of migration of places. As mentioned before, code for diffusive patterns for systems with Neumann and periodic boundary conditions are provided in the Appendix.

A deterministic diffusive system with Neumann boundary conditions requires taking care of second order effects for containers on the boundary. To implement it in sbioPN, each boundary container has to:

- receive two units from the neighbor container (even if only one unit is leaving that container).
- destroy one unit.
- send one unit to the neighbor container.

All other containers have a simpler structure, sending one unit and receiving one. For a complete treatment of this matter, see Dai's work [10].

In Figure 7.1 a one dimensional case with four interconnected containers is shown and the corresponding jumping matrix definition is presented in Table 7.1.

Each square represents a container. Green number indicates index to corresponding container. Arrows indicate jumps (first order transitions). Small number on arrow tails indicate index of jump. Gray numbers are the stoichiometry of the pre matrix (source of jump). Note: they are always ones. Red and Blue numbers are the stoichiometry of the post matrix (destination of jump).

The stochastic case is simpler, as no second order effects are considered. All the communicating containers need to send one unit to the neighbor and receive one from it. Figure 7.2 and Table 7.2 illustrate this case for the corresponding one dimensional case.

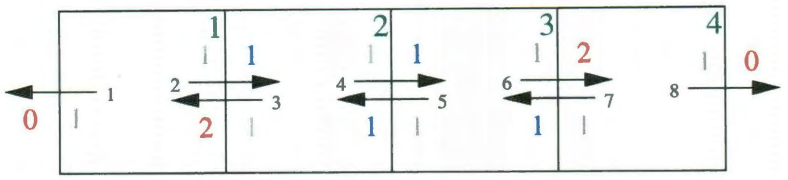


Figure 7.1 : Jumping pattern for **deterministic** diffusive system with **Neumann** boundary conditions, one dimensional case. Refer to Table 7.1 for the complete matrix definition for this case.

Jump index	Source	Destination	Stoichiometry of destination
1	1	0	0
2	1	2	1
3	2	1	2
4	2	3	1
5	3	2	1
6	3	4	2
7	4	3	1
8	4	0	0

Table 7.1 : Matrix for the jumping pattern in Figure 7.1. Only the last three columns belong to the matrix. The first column is implicit in the matrix indexing, and included for a better understanding of the relation with Figure 7.1.

For periodic boundary conditions, the same pattern can be used both for deterministic as for stochastic diffusion. See Figure 7.3 and Table 7.3 for the corresponding one dimensional case.

Diagrams for two dimensional cases are shown in Appendix F (Figure F.1 to F.3 on page 212). Following the keys from Tables 7.1 to 7.3 it is trivial but tedious to derive the corresponding jumping matrices. An algorithmical approach to generate them is advised. For example code in R see F.2 to F.4 starting on page 211.

7.4 Comparison of diffusion in a deterministic and stochastic settings

The compartment-base approach used in the previous section treats diffusion as a series of first order reactions. We have already seen in Chapter 2 that in a system where only zeroth and first order reactions happen, deterministic integration and averages of stochastic simulations should agree.

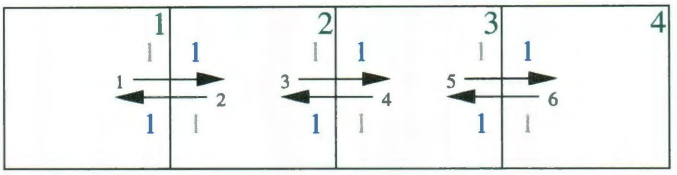


Figure 7.2 : Jumping pattern for **stochastic** diffusive system with bf Neumann boundary conditions, one dimensional case.

Jump index	Source	Destination	Stoichiometry of destination
1	1	2	1
2	2	1	1
3	2	3	1
4	3	2	1
5	3	4	1
6	4	3	1

Table 7.2 : Matrix for the jumping pattern in Figure 7.2.

To verify this in sbioPN we defined a spatial model where a zeroth order reaction fires only in the center of a 60×60 grid, creating a molecule that is allowed to diffuse. Both a deterministic integration and 100 stochastic simulations were performed. The deterministic integration is depicted in the upper panel of Figure 7.4, while the average of the stochastic runs is depicted in the bottom panel. In both cases contour lines were draw and vertical lines tangent to the contours of the deterministic case are also tangent to the corresponding level contour lines of the stochastic case, verifying that both systems evolve with the same speed and intensity.

7.5 Appropriate choice of compartment size h.

This section reviews some conclusions by Erban and Chapman (2009) [14] about the correct choice of compartment size h in cubic grids for bi-molecular reactions, both heteroreactions and homoreactions. They get to the conclusion that the compartment size has to be appropriate: it has to be a sufficient fine grid for discretization, but it cannot be arbitrarily small. There is a limit on h from below. They also note that it is recommended to use cubic grids, as it might be difficult to tell results that show a property of the system from the ones that are consequence of the non-

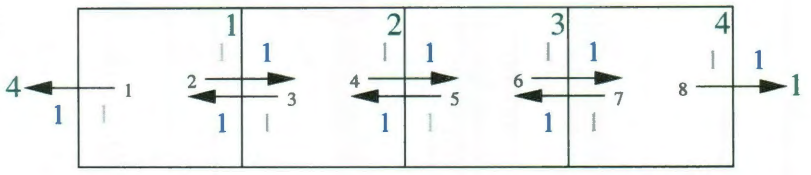


Figure 7.3 : Jumping pattern for **deterministic** and **stochastic** diffusive system with **periodic** boundary conditions, one dimensional case. Green numbers outside containers are index to destination container (on opposite side).

Jump index	Source	Destination	Stoichiometry of destination
1	1	4	1
2	1	2	1
3	2	1	1
4	2	3	1
5	3	2	1
6	3	4	1
7	4	3	1
8	4	1	1

Table 7.3 : Matrix for the jumping pattern in Figure 7.3.

uniform grid. On the other side, Engblom et al. (2009) [13] treat simulation of stochastic reaction-diffusion processes on unstructured (triangular) grids, which is a future planned addition to sbioPN.

Erban and Chapman corrections to the SSA reaction rates are summarized in the following subsections.

7.5.0.1 Heteroreactions

$$A + B \xrightarrow{k_1} \cdots \tag{7.1}$$

The usual propensity is

$$\alpha_1(t) = A(t)\, B(t) \frac{k_1}{\nu} \tag{7.2}$$

Considering that

$$h \geq h_{crit} = \beta_\infty \frac{k_1}{D_A D_B} \tag{7.3}$$

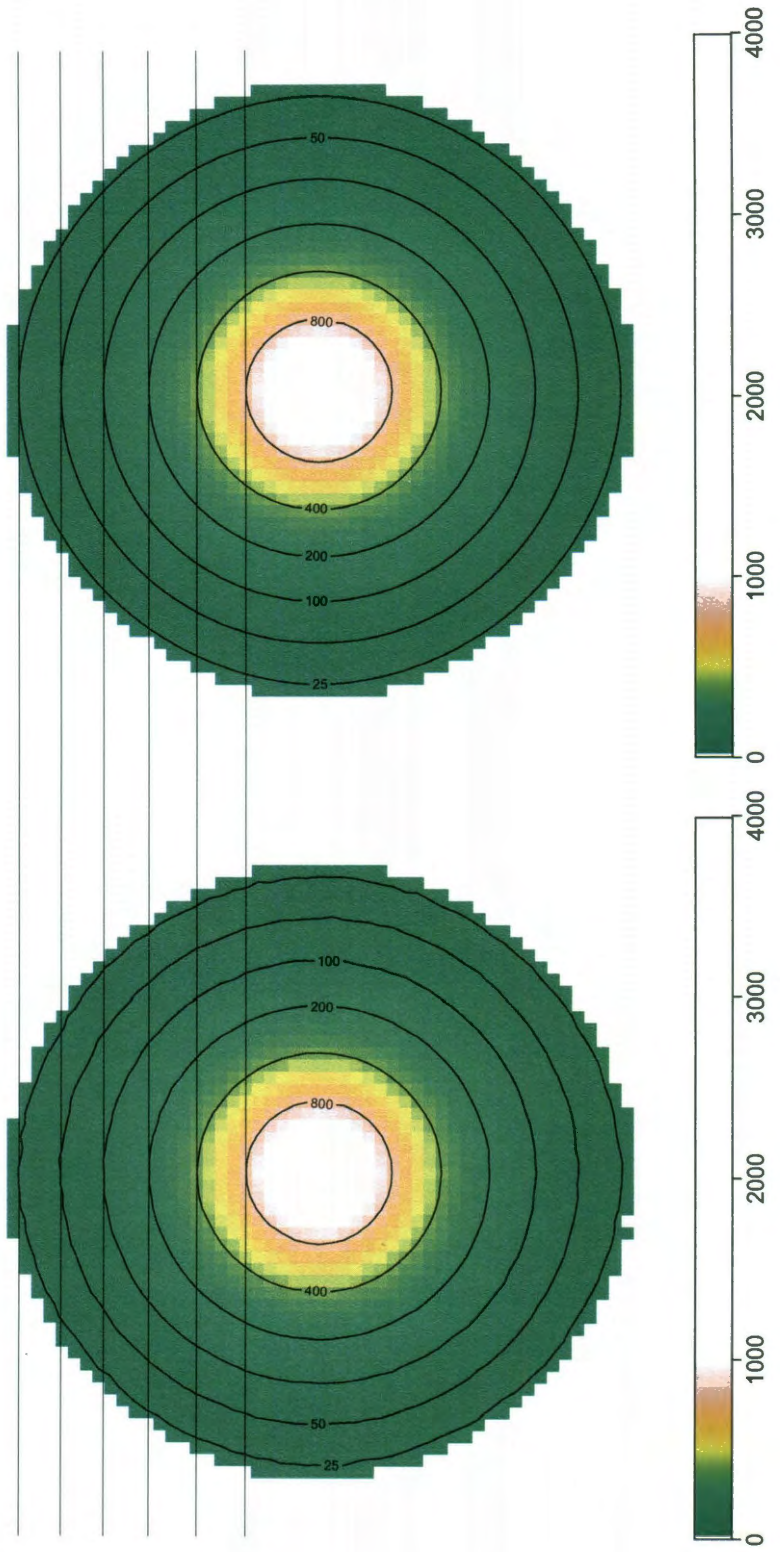


Figure 7.4 : Comparison of deterministic and stochastic diffusion. Top panel shows the deterministic run, while bottom panel shows the average of 100 stochastic runs.

where $\beta_\infty \approx 0.25272$, the proposed propensity is

$$\alpha_{ijk,1}(t) = A_{ijk}(t)\, B_{ijk}(t) \frac{(D_A + D_B)k_1}{(D_A + D_B)h^3 - \beta k_1 h^2} \tag{7.4}$$

where β is adimensional and obtained from

$$\beta = \frac{1}{2K^3} \sum_{\substack{i,j,k=0 \\ (i,j,k)\neq(0,0,0)}} \frac{1}{3 - cos(i\pi/K) - cos(j\pi/K) - cos(k\pi/K)} \tag{7.5}$$

where $K = L/h$.

If there is no information about the system (or if K is large), β_∞ is suggested.

7.5.0.2 Homoreactions

$$A + A \xrightarrow{k_1} \cdots \tag{7.6}$$

The usual propensity is

$$\alpha_1(t) = A(t)\,(A(t) - 1)\frac{k_1}{\nu} \tag{7.7}$$

NOTE: It seems it should be divided by 2

In this case, the proposed propensity is

$$\alpha_{ijk,1}(t) = A_{ijk}(t)\,(A_{ijk}(t) - 1)\frac{D_A k_1}{D_A h^3 - \beta k_1 h^2} \tag{7.8}$$

7.6 Validation of sbioPN using two examples from literature

7.6.1 Complex patterns in a simple system. Pearson 1993.

The author of this work describes a simple reaction-diffusion model with a variety of irregular spatio-temporal pattens (Turing instabilities) [55]. The system is a modification of the auto-catalytic model of glycolysis by Selkov due to Gray-Scott (original

references in [55]). The mathematical model is:

$$\frac{\partial U}{\partial t} = D_u \nabla^2 U - UV^2 + F(1-U)$$
$$\frac{\partial V}{\partial t} = D_v \nabla^2 V + UV^2 - (F+k)V$$

The system size is 2.56×2.56 (dimensionless units), with periodic boundary conditions. The initial state is defined as $U = 1, V = 0$. The 20×20 grid area in the middle is set to $U = \frac{1}{2}, V = \frac{1}{4}$ and perturbed with $\pm 1\%$ random noise to break the square symmetry.

The original system was deterministically integrated for $200,000$ time steps using a forward Euler method with time step equal to 1. The sbioPN integration is performed with a Runge-Kutta Dormand Prince 45 method [11] with adaptive step size.

The original results are found in Figure 7.5, for values of F and k as defined in the phase diagram at the top. The paper does not specify the exact values used for each case, and each pattern can be obtained by several combinations of F and k. The extracted parameters used for the sbioPN simulation are presented in Table 7.4. The corresponding results appear in Figures 7.6 and 7.7 for an early and the latest stage of the simulation.

It is not possible to reproduce exactly the same plots for the following three reasons:

- different integration technique (simple Euler method with fixed step size vs Runge Kutta Dormand Prince 45 with adaptive step-size)
- lack of specification of exact pair of parameters used for each case
- different initial random perturbation of 20×20 middle grid area

However, by comparing the end results of simulations (7.7) to Figure 7.5, a good correspondence is observed for the parameter chosen (in Table 7.4).

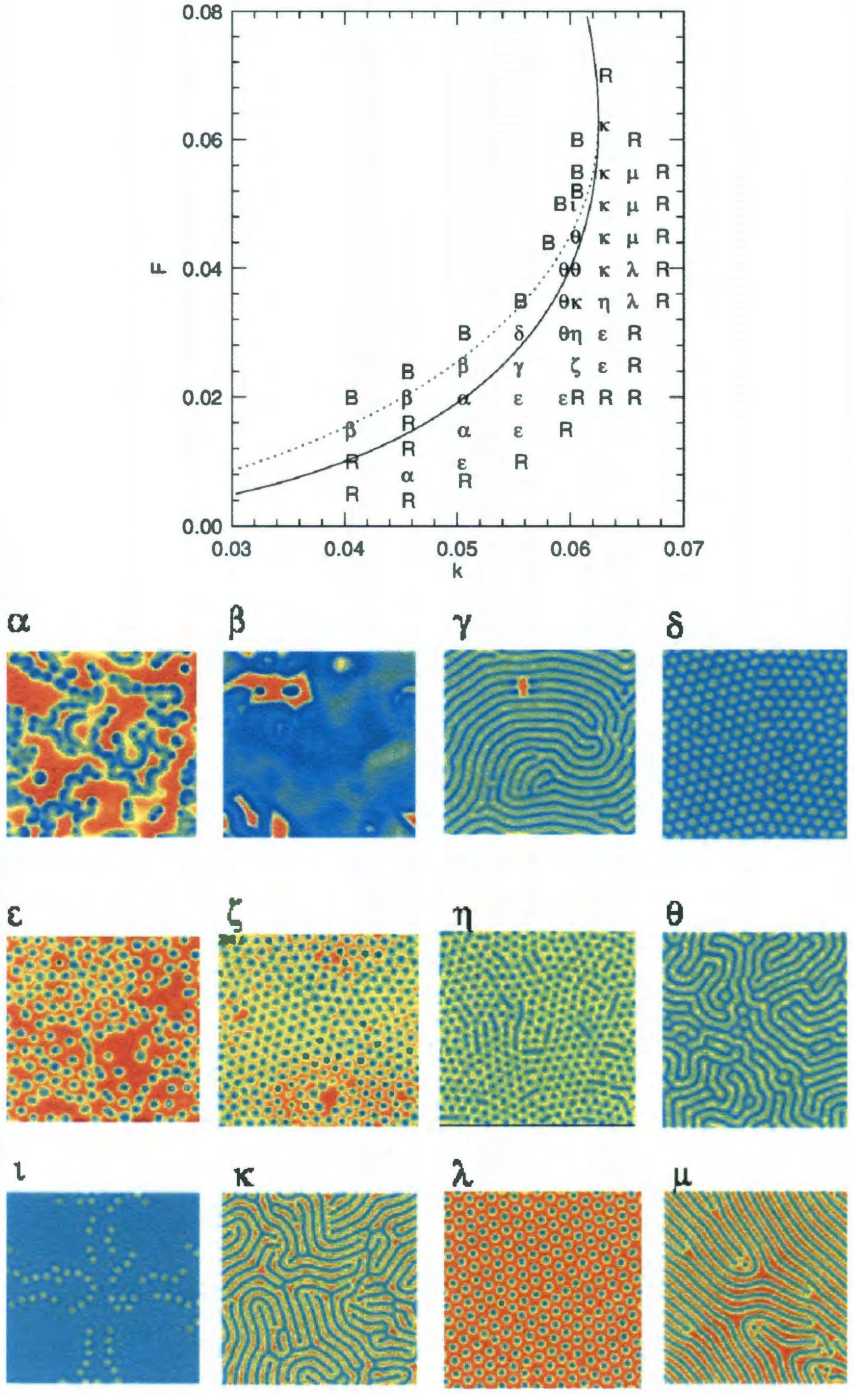


Figure 7.5 : Pearson 1993 [55]. On top: phase diagram of the reaction kinetics. Greek letters represent the location where the patterns in the figure at the bottom were found.

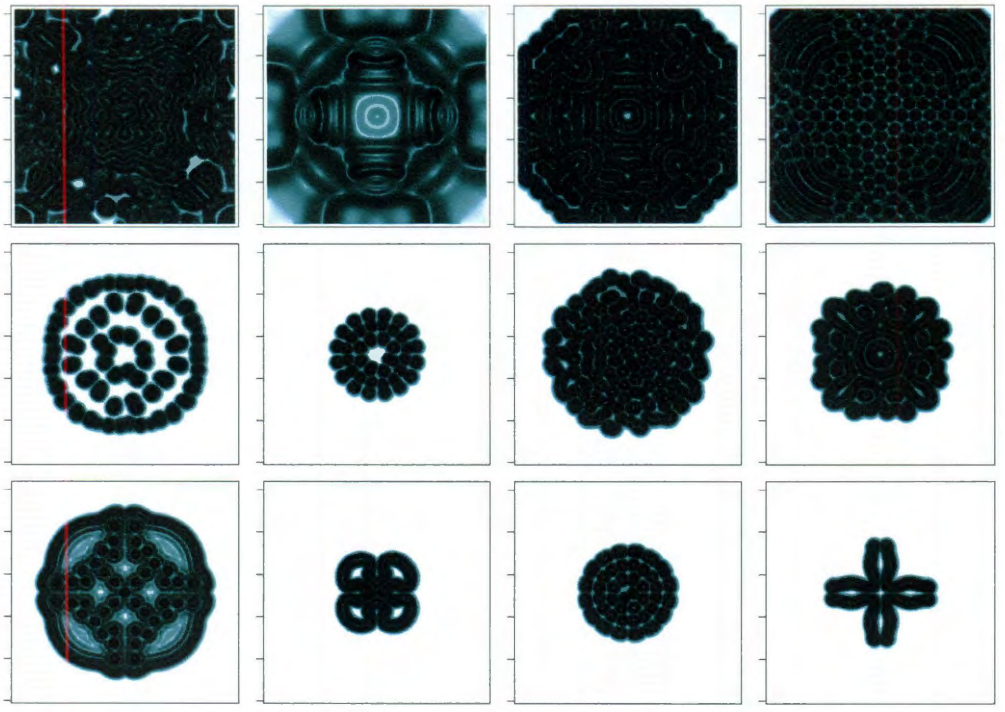


Figure 7.6 : sbioPN: Early stage of simulation.

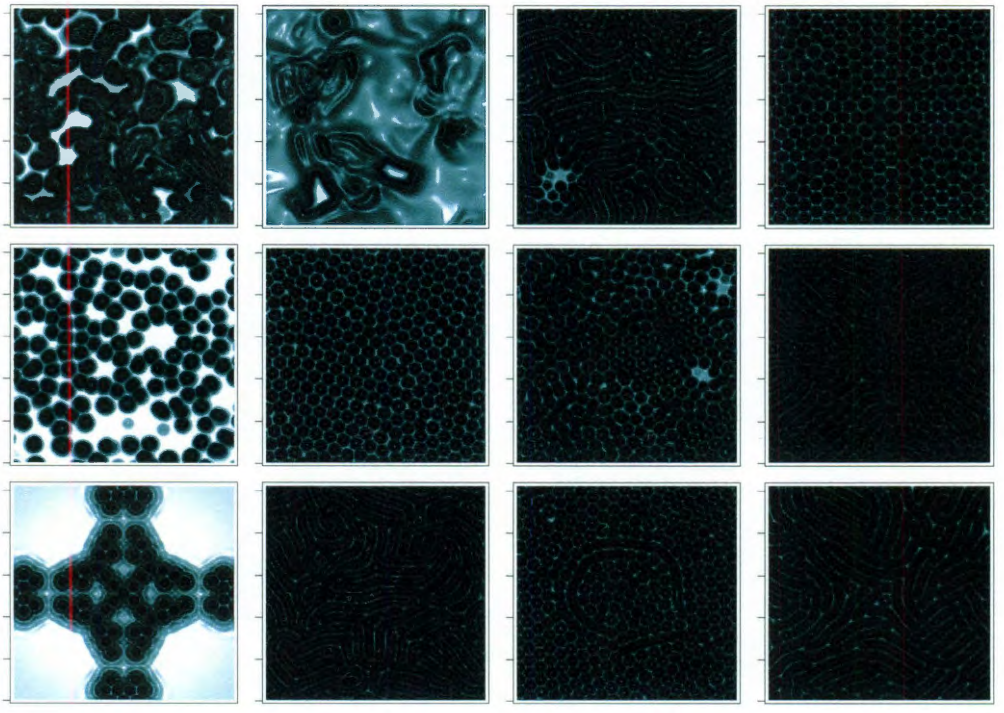


Figure 7.7 : sbioPN: End of simulation.

Case	F	k	Case	F	k
α	0.02	0.05	η	0.024	0.058
β	0.02	0.046	θ	0.03	0.06
γ	0.024	0.056	ι	0.05	0.06
δ	0.033	0.056	κ	0.044	0.063
ϵ	0.016	0.056	λ	0.038	0.064
ξ	0.033	0.064	μ	0.042	0.064

Table 7.4 : Values of F and k used in sbioPN simulation of Pearson 1993 model.

7.6.2 Okubo et al. (1989)

We present one second example from literature to illustrate that the use of sbioPN is not limited to biological networks, and that it can be used in any system in which number of elements are considered.

The authors introduce a reaction-diffusion competition model aimed at describing interactions between an exotic introduced gray squirrel and the indigenous red squirrel in Great Britain, using Lotka-Volterra type equation with isotropic diffusion [51].

The system is mathematically described by the following set of partial differential equations:

$$\begin{aligned} \frac{\partial S_1}{\partial T} &= D_1 \nabla^2 S_1 + a_1 S_1 (1 - b_1 S_1 - c_1 S_2) \\ \frac{\partial S_2}{\partial T} &= D_2 \nabla^2 S_2 + a_1 S_2 (1 - b_2 S_2 - c_2 S_1) \end{aligned} \tag{7.9}$$

where $i = 1$ for gray and $i = 2$ for red squirrels, a_i are net birth rates, $1/b_i$ are carrying capacities, c_i are competition coefficients, and D_i are diffusion coefficients. It is assumed that grays out-compete the reds, so $b_2 > c_1$ and $c_2 > b_1$.

It was used a 4.9x2.4 rectangular grid with Newman boundary conditions. The initial distribution is as follows: red squirrels are at unit normalized density, and there are 4 pockets of gray squirrels with density 0.1. Results are presented in Figure 7.8 and 7.9.

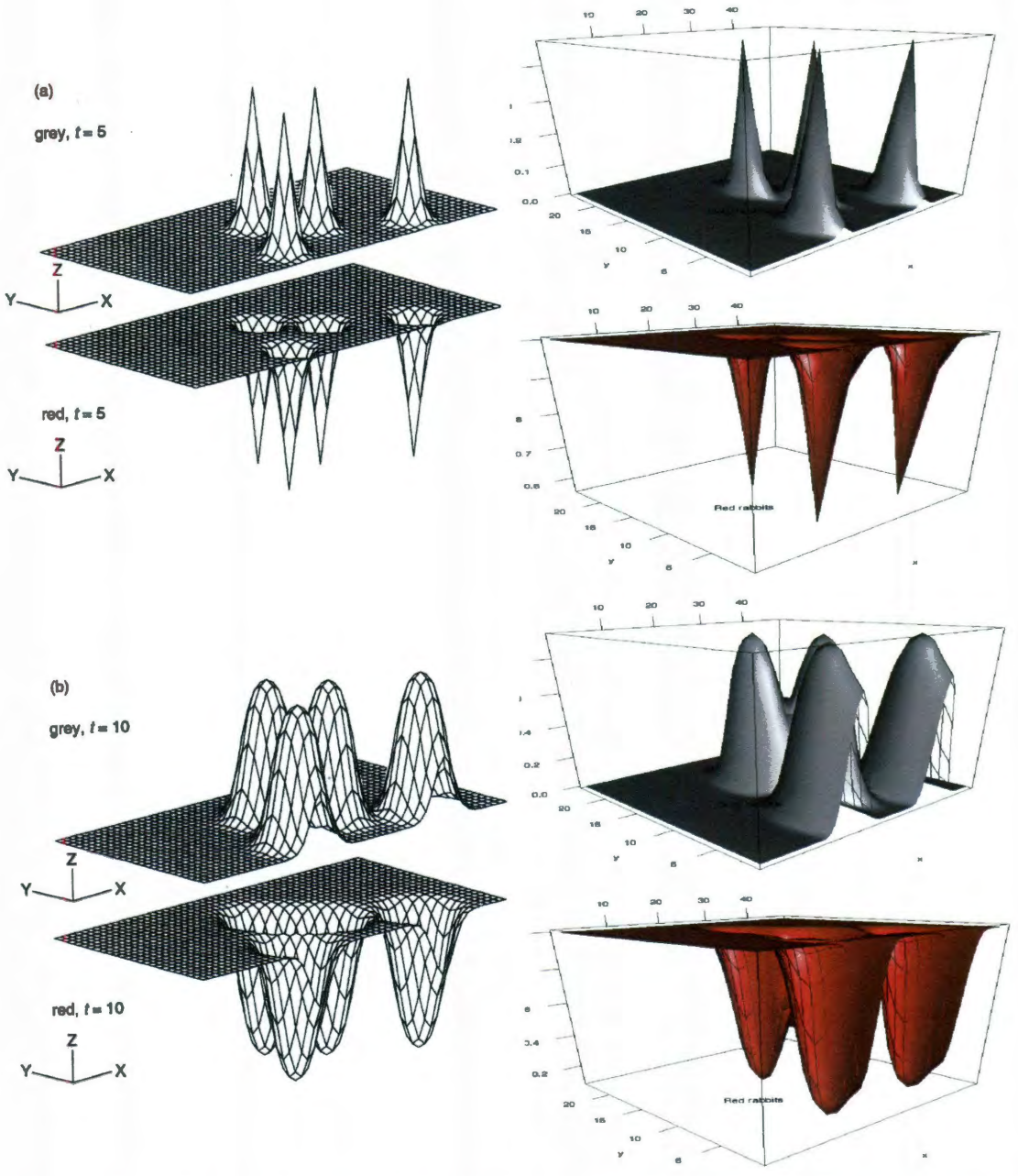


Figure 7.8 : Okubo et al. [51]. $t = 5$ and $t = 10$. On the left, results by Okubo et al. On the right, results using sbioPN.

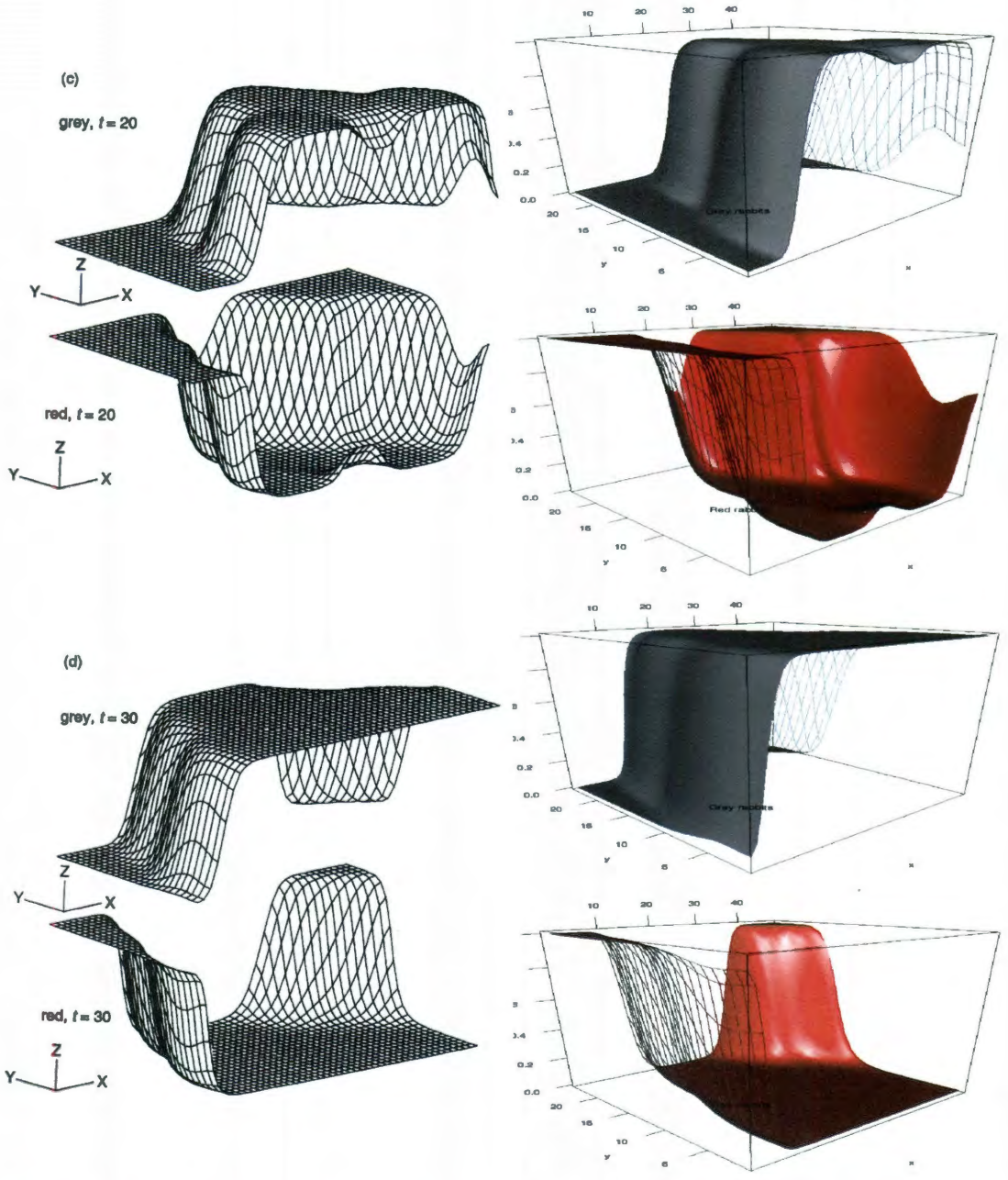


Figure 7.9 : Okubo et al. [51]. $t = 20$ and $t = 30$. On the left, results by Okubo et al. On the right, results using sbioPN.

Chapter 8

Stochastic reaction-diffusion model of early carcinogenesis

In this chapter we present the stochastic version of an original one-dimensional reaction-diffusion model of early carcinogenesis that exhibits Turing instabilities. It consists of five parts. In part one we introduce the model. In part two we describe the deterministic and stochastic approaches employed. In part three we present results of both approaches and compare them. In part four we study the behavior of the system for different rates of diffusion. In part five we treat additional material of technical nature.

8.1 Introduction

We consider the early carcinogenesis model originally proposed as a deterministic reaction-diffusion system. The model has been conceived to explore the spatial effects stemming from growth regulation of precancerous cells by diffusing growth factor molecules.

The model exhibits Turing instability producing transient spatial spikes in cell density, which might be considered a model counterpart of emerging foci of malignant cells. However, the process of diffusion of growth factor molecules is by its nature a stochastic random walk. An interesting question emerges as to whether the dynamics of the deterministic diffusion model is an approximation of the stochastic process generated by the model.

We address this question by simulations using sbioPN, presented in Chapter 7. The conclusion is that whereas single-realization dynamics of the stochastic process is very different from the behavior of the reaction diffusion system, it is becoming more

similar when averaged over a large number of realizations. The discrepancy increases as the rate of diffusion decreases.

The large-sample mean of the stochastic simulations seems to display different spatial patterns from those caused by the Turing instability in the reaction-diffusion system. Despite the differences, typical realizations of the stochastic process include spikes of cell density, which however are spread more uniformly and are less dependent of initial conditions than those produced by the reaction-diffusion system.

8.1.1 Model of early carcinogenesis

The model of a precancerous cell population is as in reference [46], which adopts elements of the models of a homogeneous population of cells of one type previously proposed in references [43,45]. The present model is based on the following hypotheses:

- Precancerous cells c, existing in a spatial domain, proliferate at a rate $a(b,c)$, which is reduced by cell crowding but enhanced in a paracrine manner by a hypothetical bio-molecular growth factor b bound to cells.

- Pre-cancerous cells are supplied at a constant rate μ by mutation of normal cells.

- Free growth factor g is secreted by the cells at the rate $\kappa(c)$, then it diffuses among cells with diffusion constant $1/\gamma$, and binds to cell membrane receptors at a rate $\alpha(c)$, becoming the bound factor b. It then dissociates at a rate d, returning to the free factor pool.

- Free and bound growth factor particles decay at rates d_g and d_b, respectively.

Discussion of possible geometries for the spatial variable x can be found in previous papers [43,44]. One natural geometry is that of a line of cells, occupying the interval

$x \in [0,1]$. There are three substances distributed over the line's length: Cells and free and bound growth factor molecules, with densities $c(x,t)$, $g(x,t)$ and $b(x,t)$, respectively. The resulting equations are as follows:

$$\frac{\partial c}{\partial t} = (a(b,c) - d_c)c + \mu, \tag{8.1}$$

$$\frac{\partial b}{\partial t} = \alpha(c)g - d_b b - db, \tag{8.2}$$

$$\frac{\partial g}{\partial t} = \frac{1}{\gamma}\Delta_x g - \alpha(c)g - d_g g + \kappa(c) + db, \tag{8.3}$$

with homogeneous Neumann (zero flux) boundary conditions for g

$$\partial_x g(0,t) = \partial_x g(1,t) = 0 \tag{8.4}$$

The kinetics were derived from the stochastic model describing the transitions between different states of the growth factor molecules [43, 44]. Diffusion equation is a macroscopic approximation of the microscopic process of growth factor binding to membrane receptors, under homogeneity hypotheses; it has been derived in [47]. Coefficient $1/\gamma$ is a composite parameter including the diffusion constant and scaling parameters, $\gamma = 1/d$. Proliferation rate has the Hill function form

$$a(c,b) = \frac{a_1(b/c)^m}{1 + (b/c)^m} \tag{8.5}$$

where $a_1 = (2p-1)a_0$ and p is the efficiency of divisions. We will consider the special case $m = 1$. Production of free growth factor by cells has the Michaelis-Menten form

$$\kappa(c) = \frac{\kappa c}{1 + c} \tag{8.6}$$

The form of growth factor binding rate $\alpha(c)$ has been obtained from conditions for diffusion-driven instability investigated in ref. [44]. It assumes the form,

$$\alpha(c) = \alpha_1 c^{s+1} \qquad s > 0 \tag{8.7}$$

i.e., the process of binding free growth factor particles to cells is super-linear.

The differential equations where transformed to the corresponding set of coupled chemical reactions. The set as well as considerations about diffusion and boundary conditions are described in Section 8.6.

8.2 Deterministic reaction-diffusion system

8.2.1 Spatially homogeneous steady state

From reference [46], the spatially homogeneous steady state of pre-cancerous cells ($\bar{c}$) is the solution to:

$$\kappa \frac{c^2}{d_g \frac{d_b+d}{\alpha}(1+c) + \kappa c^2 + d_b c^2(1+c)} + \frac{\mu}{a_1 c} - \frac{d_c}{a_1} = 0 \tag{8.8}$$

The corresponding expression for bound growth factor is:

$$\bar{b} = \kappa \frac{\bar{c}}{1+\bar{c}} \frac{\bar{c}^2}{d_b \bar{c}^2 + d_g \frac{d_b+d}{\alpha}} \tag{8.9}$$

and the one for free growth factor is:

$$\bar{g} = \frac{d_b + d}{\alpha \bar{c}^2} \bar{b} \tag{8.10}$$

The resulting values using parameters specified in Table 8.1 are $\bar{c} = 5.86$, $\bar{b} = 8.07$, and $\bar{g} = 0.47$. Numerical integration of the deterministic system agrees with these results, which are presented in the upper panel of Figure 8.3.

Parameter	Value
a_1	$1/12$
α	10^{-1}
κ	1
s	1
d_c	5×10^{-2}
μ	10^{-2}
$d = d_b = d_g$	10^{-1}

Table 8.1 : Values of the parameters used in all studied systems.

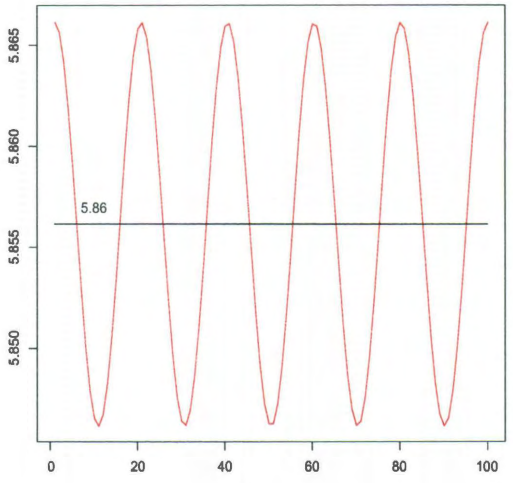


Figure 8.1 : Deterministic reaction-diffusion system. Red line shows perturbation of $\bar{c}$ (black line), according to Equation 8.11, with $\epsilon = 10^{-2}$ and $n = 5$.

8.2.2 Perturbation of spatially homogeneous steady state

To break the spatially homogeneous equilibrium, a cosinusoidal perturbation is applied to $\bar{c}$ using the expression

$$c_0 = \bar{c} + \epsilon \cos(2n\pi x), \quad 0 \leq x \leq 1 \tag{8.11}$$

where $\epsilon = 10^{-2}$ is the amplitude used [46], and n is the number of peaks. The case $n = 5$ is illustrated in Figure 8.1.

8.2.3 Numerical integration of perturbed system

Ten different grid densities, five different γ values ($\gamma = 1; 10; 100; 1000;$ and $10,000$), and five different n values (1 to 5) were considered. As a consequence, we studied a total of two hundred and fifty perturbed deterministic reaction-diffusion systems. In all cases, including the stochastic simulations, the parameter values used were the same and are summarized in Table 8.1.

Each perturbed deterministic system was numerically integrated for 2000 time units, and the values at 2000 time units, rounded to the nearest integer, were used as the initial conditions for stochastic simulations.

We refer as "main case" the one having the following parameter values: $\gamma = 100$, and $n = 5$, as this is the case from reference [46] that motivated our study and it was the most studied case. It shares the same characteristics than the rest of the cases with equal γ, but its behavior is different when compared to cases with other γ values. Those cases will be treated in Section 8.4. We will start by only considering a grid density of 100 nodes. We will later (Section 8.7) analyze the change of behavior of the main case as the grid density changes.

The time evolution of the main case is presented in the upper panels of Figure 8.4. The left column shows the time evolution of pre-cancerous cells, and the right column the corresponding time evolution of free growth factor. Bound growth factor is not included in the plots as it follows the exact shape of the pre-cancerous cell plots in a different scale.

As a result of the integration, this case shows, at 2000 time units, a symmetric configuration of four full spikes and two "half spikes" at the boundaries. This case was also integrated for 4000 time units, not showing a change in behavior.

8.3 Stochastic reaction-diffusion system

To study the behavior of the stochastic system, simulations with three different initial conditions were performed:

1. with all the initial quantities set to zero,
2. with the spatially homogeneous steady states rounded to the nearest integer,
3. with the end values of the deterministic perturbed systems rounded to the nearest integer. Two hundred and fifty stochastic simulations have been performed with this kind of initial conditions, i.e., the same number as in the deterministic case.

Rounding to the nearest integer is an unavoidable requirement for stochastic simulation. Potential implications and further studies are deferred to Section 8.8.

8.3.1 Single runs

1000 runs were performed for each stochastic simulation. In this section we show sample realizations using two of the initial conditions specified above (the last one corresponding to the main case). We only show the evolution of pre-cancerous cells. Results for bound growth factor, albeit not exactly identical, are very similar to those for pre-cancerous cells with a rescaling factor, and runs of free growth factor do not show any discernible structure, and are deferred to figures of mean values.

Left panel of Figure 8.2 depicts a typical run with zero initial conditions. Spikes appear in a mostly uniform way along the x axis which last for a variable amount of time, being replaced by spikes formed in different places. The spikes do not seem to form in any preferential spot.

Stochastic simulations using as initial conditions the spatially homogeneous steady state values of the deterministic system rounded to the nearest integer (not shown) rapidly evolve to a similar behavior as the corresponding to zero initial values.

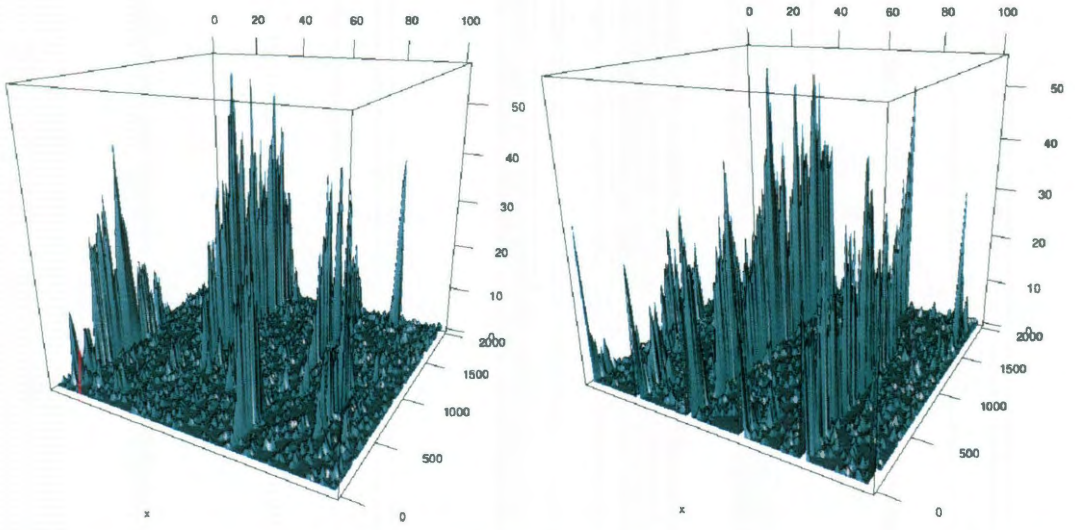


Figure 8.2 : Single runs of the stochastic reaction-diffusion system. Left panel depicts a run with zero initial conditions, while the right panel depicts a run with initial condition set to the end values of the deterministic system rounded to nearest integer. In both cases $\gamma = 100$ was used.

The right panel of Figure 8.2 depicts a typical stochastic runs with initial conditions set to the values of solutions of the perturbed main case at 2000 time units, rounded to the nearest integer. The initial conditions in this case have four full spikes and two half spikes at the boundaries. Results show that the stochastic system does not preserve all the spikes. Most of them disappear rapidly, and no more than two spikes coexist in some runs.

8.3.2 Averages of 1000 runs

Figure 8.3 shows a comparison between deterministic and stochastic systems before perturbation of the deterministic system. The average of the number of pre-cancerous cells in the stochastic simulation using zero initial conditions is significantly lower than the result of the deterministic simulation (5.86 compared to values below 1). This behavior is also mimicked by bound growth factor (not shown). In the case of free growth factor, instead, both cases seem to converge to values close to the deterministic one.

To further analyze the cause of this behavior, we performed deterministic and

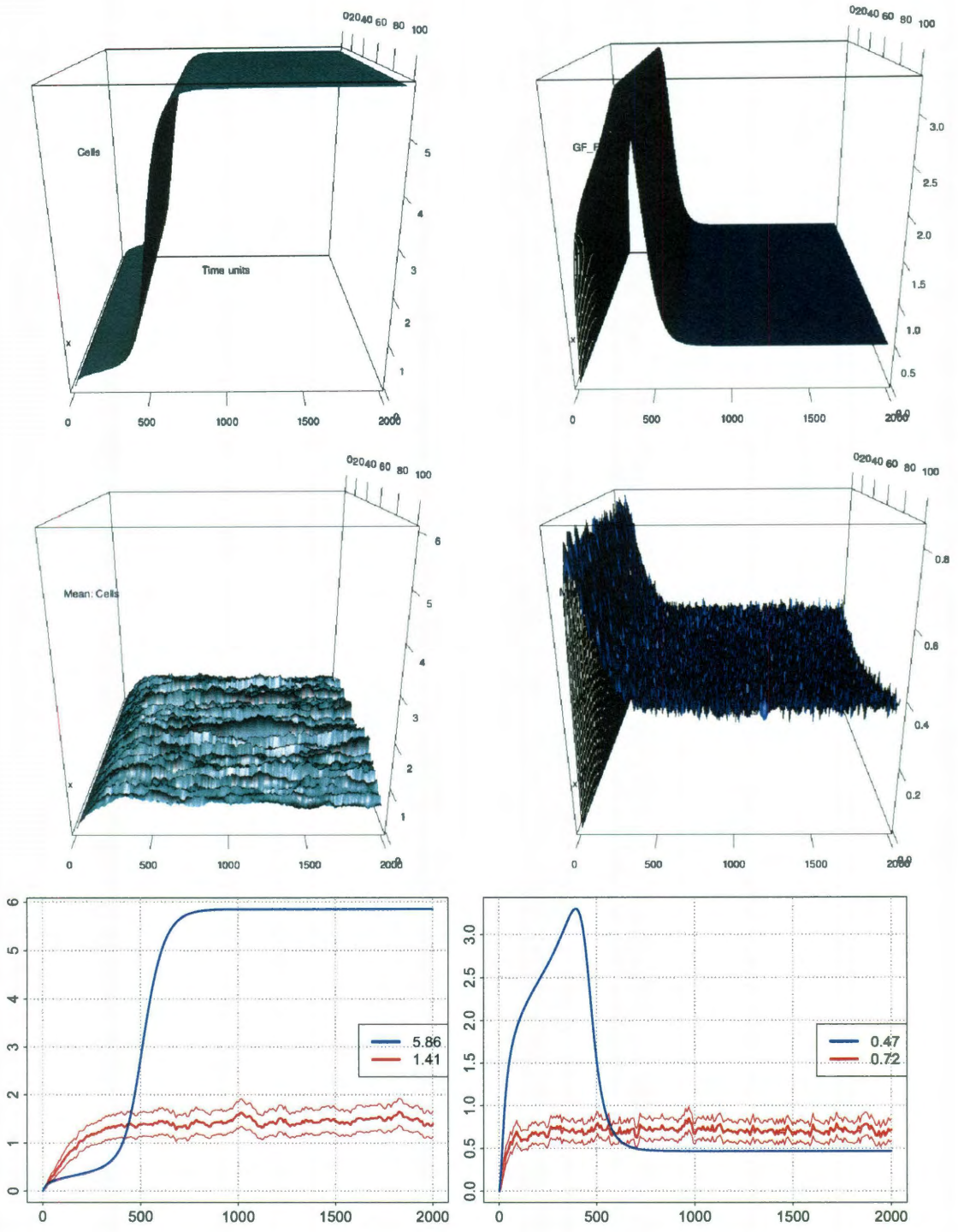


Figure 8.3 : Deterministic/stochastic reaction-diffusion and non diffusive systems comparison. Zero initial conditions are used. Depicted are, on the left, pre-cancerous cells and, on the right, free growth factor. First row shows deterministic averages, while second row averages of 1000 stochastic runs. Third row shows the non diffusive system, where blue lines correspond to deterministic mean and red lines correspond to averages of 1000 stochastic runs. The values in the legends correspond to time 2000.

stochastic simulations for the associated non diffusive model, and we present the results in the lower panel of Figure 8.3. The deterministic and stochastic mean values of pre-cancerous cells differ by almost four folds, in a similar way as what is observed in the spatial case. The deterministic mean values for free growth factor are identical for spatial and non-spatial cases.

However, the stochastic mean of free growth factor differs with respect to the non-spatial stochastic mean, both in initial shape as in the equilibrium values achieved. The first rapidly increases and decreases at the initial time units, reaching a stable plateau at about 0.4 at time 500. The second, instead, rapidly grows until reaching a plateau at about 0.7 in a very short time.

The initial increase and decrease in mean values for the spatial case may be related to the diffusive characteristic of free growth factor: as it is not produced equally along the x axis, the observed effect may be related to an initial imbalance that is compensated as time progresses. However, we do not have and explanation that justifies why the diffusive and non diffusive systems achieve a different equilibrium level.

Figure 8.4 shows a comparison of the perturbed deterministic main case system and the averages of 1000 runs of the stochastic system, assuming as initial conditions the end values of the deterministic system rounded to nearest integer. The initial-value spikes of the pre-cancerous cells rapidly lower to about two thirds of the initial value, and then continue to decrease at a lower rate. Based on this alone, it is not clear if after 2000 time units the spike values continue to decrease until they reach a limit value or until reaching the same level as in Figure 8.3. Vanishing of the half spikes at the boundaries suggests the second option.

To address this issue a new series of 1000 runs was performed for the main case with time ranging up to 8000 units. The results are depicted in Figure 8.5 which show that, in the long run, the averaged spikes disappear from the stochastic system.

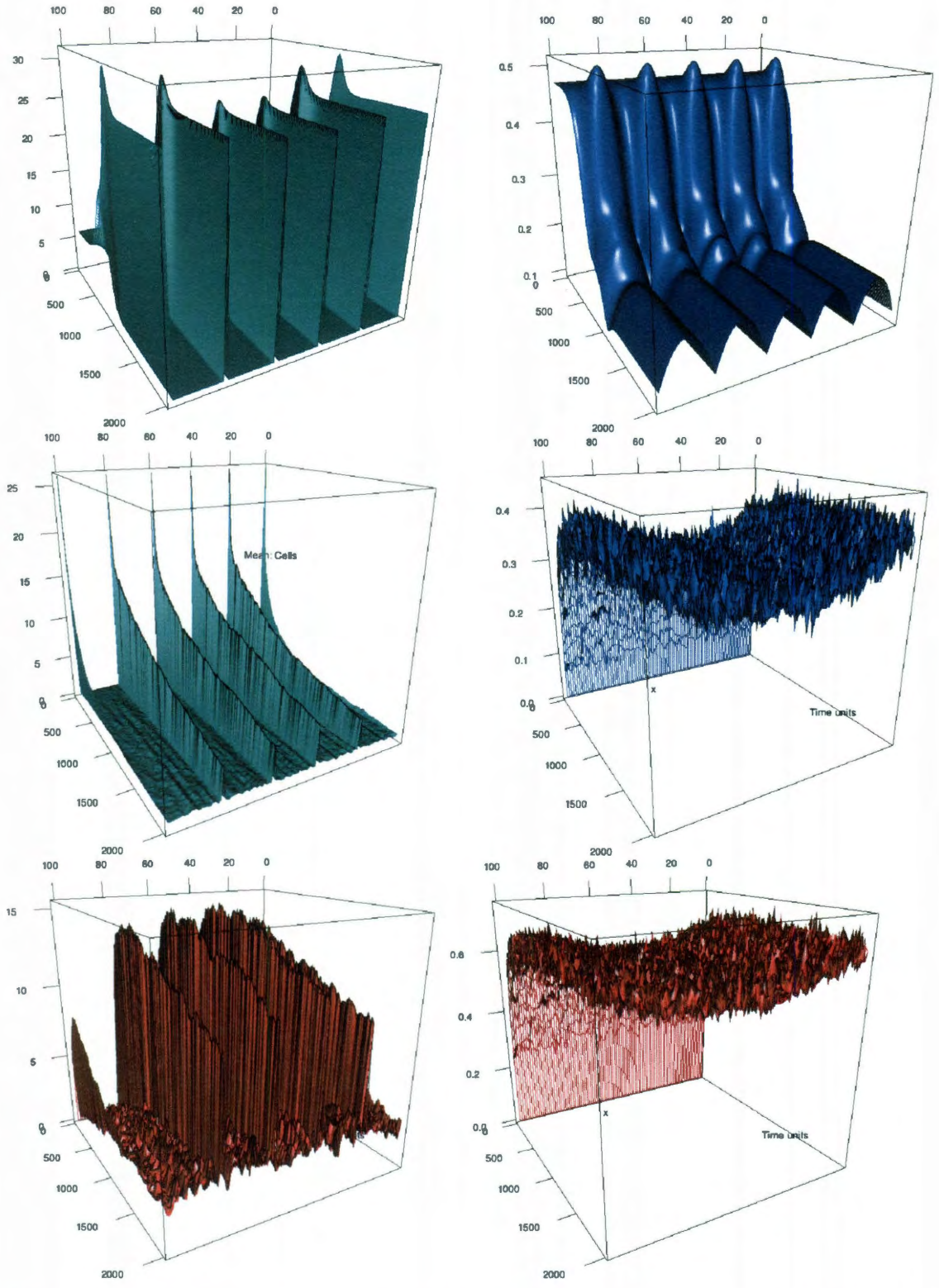


Figure 8.4 : Deterministic/stochastic reaction-diffusion systems comparison, after perturbation. On the left are depicted pre-cancerous cells, while in the right are depicted free growth factor molecules. First row corresponds to deterministic evolution after perturbation. Second and third rows correspond to averages and standard deviation, respectively, of 1000 stochastic runs with initial conditions set to the end values of the deterministic system rounded to nearest integer.

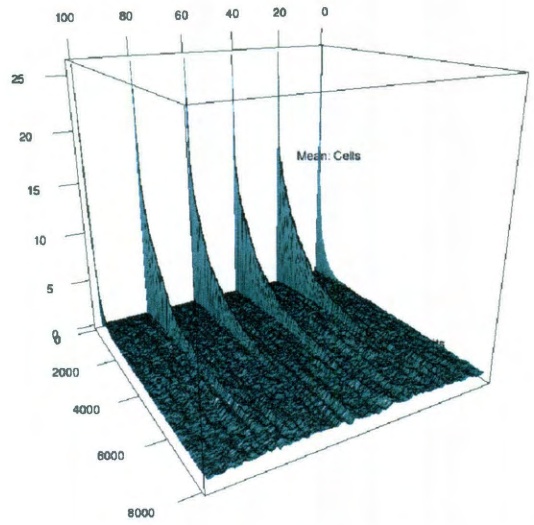


Figure 8.5 : Stochastic reaction-diffusion system studied for 8,000 time units. Depicted are averages of 1000 stochastic runs of pre-cancerous cells, with initial conditions set to the end values of the deterministic system rounded to nearest integer.

8.4 Behavior of the system for different rates of diffusion

Let us remind that in the so-called main case, the parameter values assumed were $n = 5$ and $\gamma = 100$. In this section we will present a comparison of deterministic and stochastic results obtained when studying the case $n = 3$ for the range of γ values that we studied: $\gamma = 1; 10; 100; 1000;$ and $10,000$. As γ is the reciprocal of the diffusion constant, lower values of γ imply higher diffusion rates.

We have chosen $n = 3$ because the associated deterministic systems exhibits two full and two half spikes for γ assuming values 1 and 10, providing good initial values for the stochastic simulations. Deterministic cases associated to other n values provide less interesting initial values for these γ values (most of them only show the half spikes at the boundaries for at least one of the γ values, as it happens for the main case). Other parameter combinations are not shown because they lead to the same conclusions as the case of $n = 3$.

The results, presented in Figures 8.6 and 8.7 for increasing values of γ, are orga-

nized as follows: each figure shows two different values of γ, one on the left column and one on the right column. Top panels show the deterministic simulations of the corresponding perturbed system. Middle panels show the average of 1000 stochastic runs. Bottom panels show one representative stochastic run.

Interestingly, it seems that the stochastic system better preserves the averaged spikes of the pre-cancerous cells and bound growth factor (not shown), if lower values of γ are assumed, or, equivalently, if diffusion rate increases. It even seems to be able to indefinitely preserve the average spike height constant in time for $\gamma = 1$ (left column of Figure 8.6). Single runs show that one of the original spikes is maintained through completion of simulation, while the rest disappear rapidly. By observing the plot of the averages, it seems that the internal full spikes are preserved with greater probability with respect to boundary ones.

In the right column of Figure 8.6 ($\gamma = 10$) a similar behavior can be observed for the internal but not for the boundary spikes. Single runs show that the boundary spikes disappear rapidly while one of the internal spikes is maintained through completion of simulation.

The left column of Figure 8.7 depicts the case using $\gamma = 100$ that, as expected, shows behavior similar to that observed for the main case for the average values (Figure 8.4): a marked drop from the initial values followed by a gradual decay. The same can be said about the similarities found among single runs with respect to the ones of the main system (Figure 8.2).

The right column of Figure 8.7 depicts how the deterministic system becomes numerically unstable for $\gamma = 1000$. In contrast, the stochastic system, exhibits the type of behavior observed in Figure 8.3.

Finally, for $\gamma = 10,000$ (not shown) both the deterministic and stochastic systems revert to the equilibrium state.

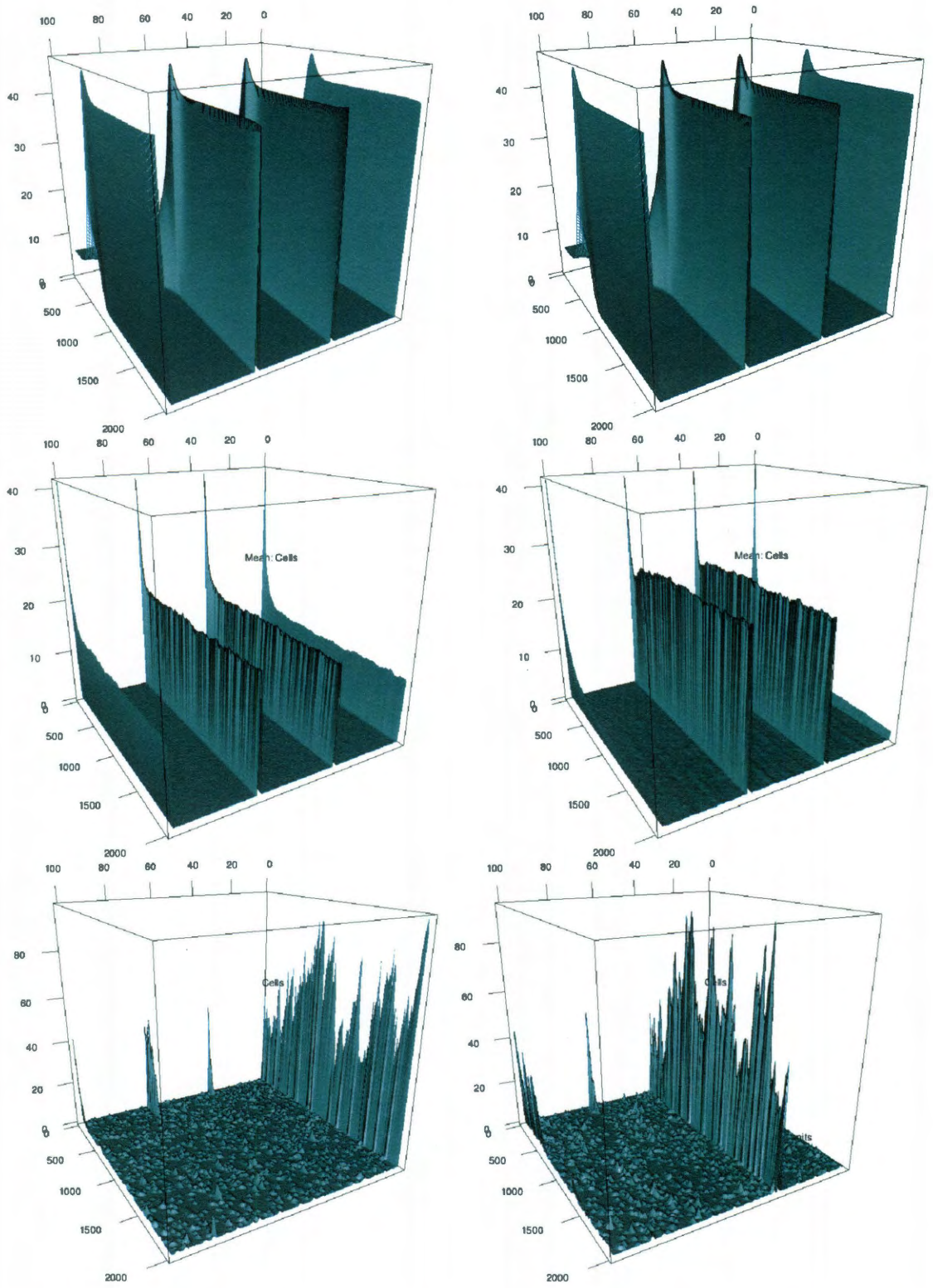


Figure 8.6 : Deterministic/stochastic comparison for different values of parameter γ. First row depicts evolutions of perturbed deterministic systems. Second row depicts averages of 1000 stochastic runs. Third row depicts single stochastic runs. The stochastic simulations had initial condition set to the ending values of the deterministic system rounded to nearest integer. All cases correspond to $n = 3$. Results for $\gamma = 1$ are on the left, while results for $\gamma = 10$ are on the right.

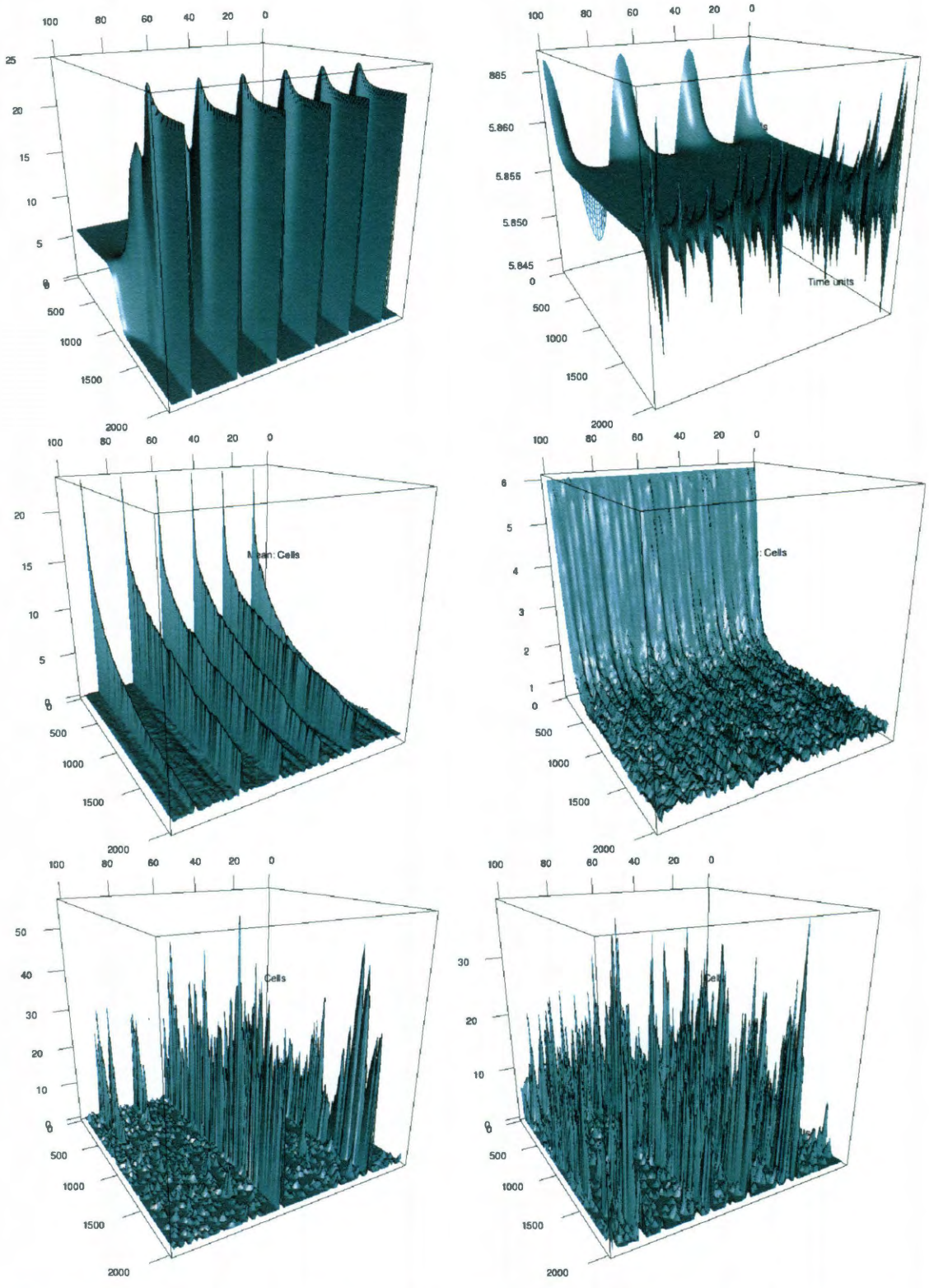


Figure 8.7 : Deterministic/stochastic comparison for different values of parameter γ. Continuation. First row depicts evolutions of perturbed deterministic systems. Second row depicts averages of 1000 stochastic runs. Third row depicts single stochastic runs. The stochastic simulations had initial condition set to the ending values of the deterministic system rounded to nearest integer. All cases correspond to $n = 3$. Results for $\gamma = 100$ are on the left, while results for $\gamma = 1000$ are on the right.

8.5 Additional material of technical nature

We will treat now arguments of technical nature that can be skipped without loosing general understanding of the topics treated in this body of work (next chapter starts on page 127).

The additional material is organized as follows:

- We present the chemical reactions that describe the system of differential equation presented in equations 8.1 to 8.3, as well as the treatment of diffusion and boundary conditions (Section 8.6 starting on page 116).
- We examine the consequences of modifying the grid density (Section 8.7 starting on page 119).
- We analyze potential consequences of rounding to nearest integer the initial conditions of the stochastic system (Section 8.8 starting on page 123).

8.6 Modeling the deterministic and stochastic reaction-diffusion systems

To have a consistent definition for the deterministic and stochastic systems, the original differential equations were transformed into the corresponding coupled chemical reactions. The description of each coupled reaction is as follows:

- Proliferation of pre-cancerous cells:

$$c + b \xrightarrow{a_1 bc/(b+c)} 2c + b \tag{8.12}$$

- Pre-cancerous cell death:

$$c \xrightarrow{d_c c} \otimes \tag{8.13}$$

- Mutation of normal cells into pre-cancerous ones:

$$\otimes \xrightarrow{\mu} c \tag{8.14}$$

- Binding of free growth factor to pre-cancerous cells:

$$c + g \xrightarrow{\alpha c^{s+1} g} c + b \tag{8.15}$$

- Degradation of bound growth factor:

$$b \xrightarrow{d_b b} \otimes \tag{8.16}$$

- Dissociation of bound growth factor from pre-cancerous cell, returning to the pool of free growth factor:

$$b \xrightarrow{db} g \tag{8.17}$$

- Degradation of free growth factor:

$$g \xrightarrow{d_g g} \otimes \tag{8.18}$$

- Secretion of free growth factor by pre-cancerous cells:

$$c \xrightarrow{\kappa c/(1+c)} c + g \tag{8.19}$$

As not all the reaction propensities are of mass-action nature, in this case the quantities appearing above the arrows represent the reaction propensities corresponding to each reaction, instead of the reaction constants.

8.7 Consequences of varying the grid density

8.7.1 Deterministic system

Equations 8.12 to 8.19 are only valid for a grid density of 100 nodes. To be able to analyze other grid densities, some of the reaction propensities need to be modified to a more general form:

- The propensity of Equation 8.14 is replaced by:

$$\mu \frac{100}{nodes} \tag{8.27}$$

- The propensity for Equation 8.15 is replaced by:

$$\alpha \left(\frac{100}{nodes} \right)^{-(s+1)} c^{s+1} g \tag{8.28}$$

- The propensity for Equation 8.19 is replaced by:

$$\kappa c \frac{100}{nodes} \left(\frac{100}{nodes} + c \right)^{-1} \tag{8.29}$$

where *nodes* is the number of nodes of the interval $0 \leq x \leq 1$.

Figure 8.8 shows cross sections of time evolution of pre-cancerous cells in the perturbed deterministic system for different number of nodes: 25, 50, 100, 200, and 400. The main case (100 nodes) is presented in the middle. Orange lines that look like horizontal straight lines are the cosinusoidal perturbations of the corresponding spatially homogeneous steady state (that are number above orange lines). The perturbation is better appreciated in Figure 8.1.

The spatially homogeneous steady states double if the number of nodes are halved, and the opposite holds true if the number of nodes are doubled. Values at 2000 time units (red lines) do not follow the same behavior. Spikes seem to decrease following a

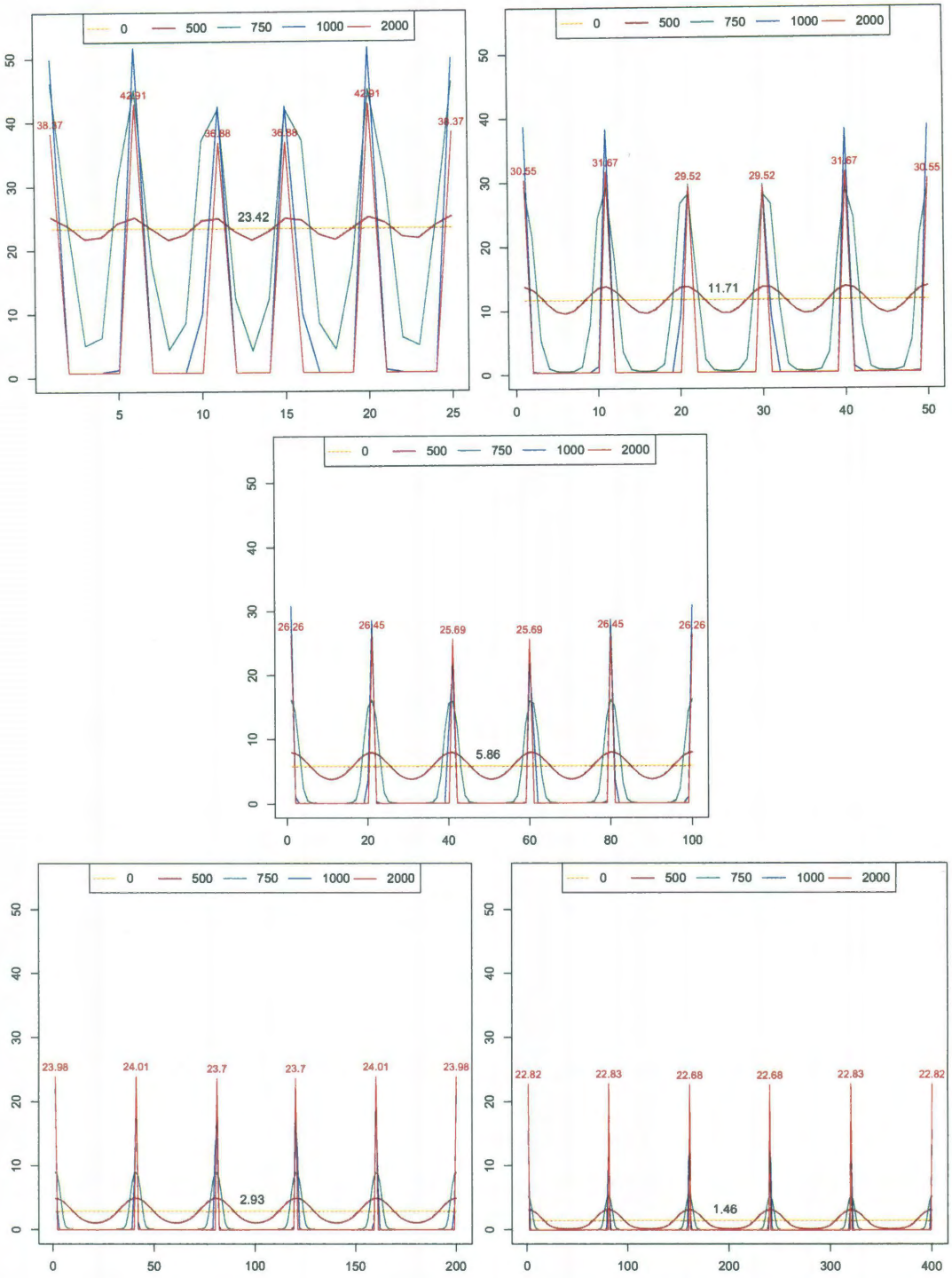


Figure 8.8 : Cross sections of evolution of perturbed deterministic systems for different grid densities. Depicted are pre-cancerous cells. Horizontal axis shows number of grid nodes of interval $0 \leq x \leq 1$.

shifted exponential law to what appear to be a limiting value, as the number of nodes increase. In all cases, spikes form in one node. A potential misleading interpretation of rescaling the results is treated in Section 8.7.2.

The same behavior and conclusion can be drawn for bound growth factor. Results are not shown because they look exactly the same as a rescaled version of pre-cancerous cells.

Free growth factor follows a more expected behavior, and the results are presented in Appendix D (Figure D.1 on page 204). Values tend to double as number of nodes half, and vice versa. Increased number of nodes favor smoother results.

8.7.2 Rescaling the results of deterministic system to match spatially homogeneous steady state values

A potentially misleading presentation of Figure 8.9 depicts the results of simulations of the deterministic system for different grid densities rescaled so that all the plots show the same spatially homogeneous steady state. This way of presentation may suggest the interpretation that, as the density of the grid increases, the spikes diverge toward infinity (becoming "Diract deltas"). As Figure 8.8 shows, this interpretation is incorrect. The effect is caused by rescaling and the fact that, numerically, the spikes only occur in one voxel, no matter how small the voxel is.

In the case of free growth factor plots, which are smoother, this artifact does not occur, as presented in Appendix D (Figure D.2 on page 205).

8.7.3 Stochastic system

Simulations of the main case were also performed for the same number of nodes presented in the previous subsection, using as initial conditions the end values of the corresponding perturbed deterministic systems rounded to the nearest integer.

Results for pre-cancerous cells are presented in Figure 8.10, which seems to lead to

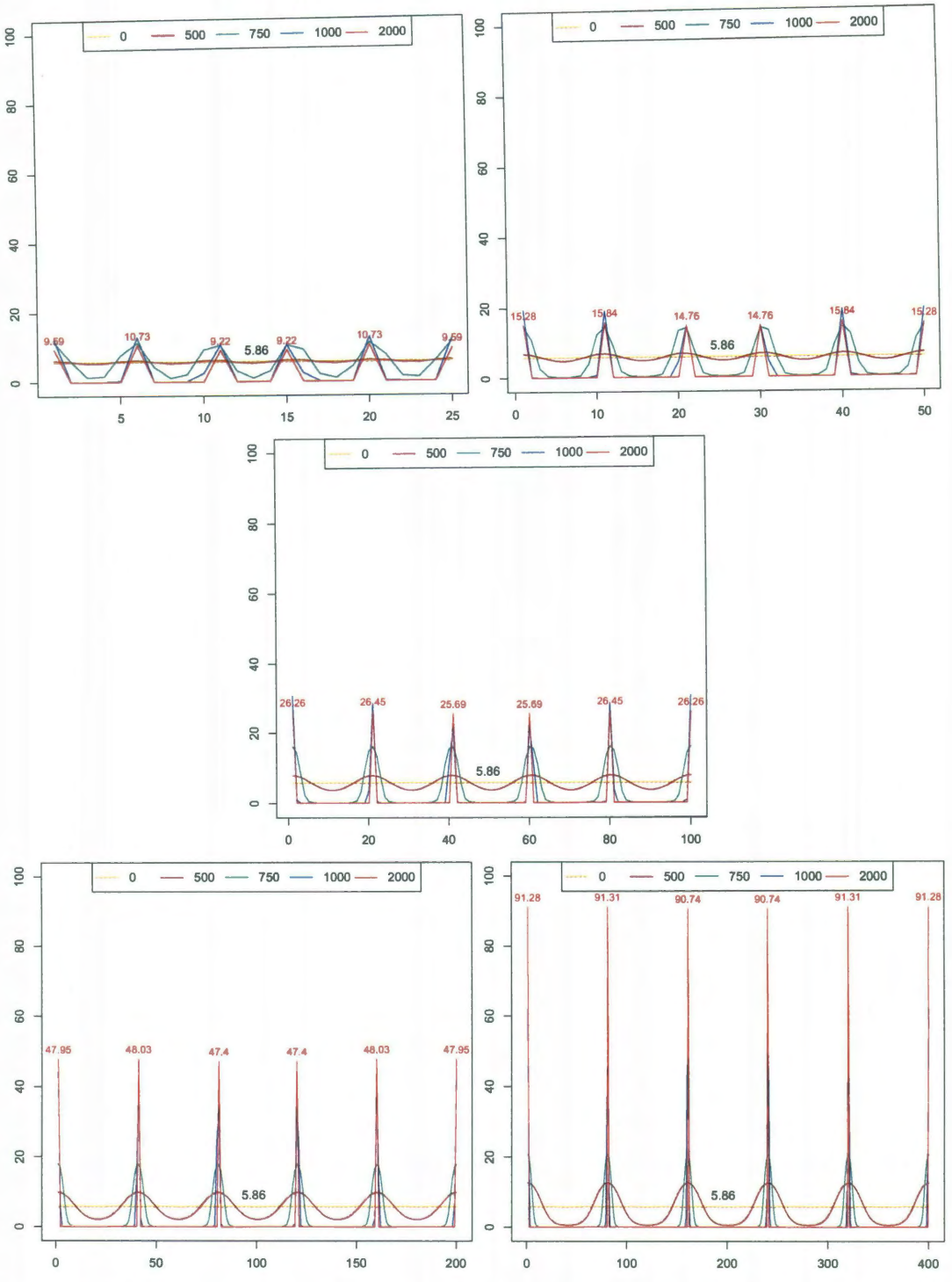


Figure 8.9 : Cross sections of evolution of perturbed deterministic systems for different grid densities, rescaled to show the same spatially homogeneous steady state as the main case (100 grid nodes). Depicted are pre-cancerous cells. Horizontal axis shows number of grid nodes of interval $0 \leq x \leq 1$.

the conclusion that the stochastic system sustains better the level of averaged spikes for coarser grids, and the opposite for finer grids.

Results for free growth factor are presented in Appendix D (Figure D.3 on page 206). Lower number of nodes convey cross-sectional structure to the averaged values, which is increasingly lost as the number of nodes increase.

8.8 Analysis of potential consequences of rounding to nearest integer the initial conditions of the stochastic system

This section studies eventual consequences of rounding to the nearest integer the results of the perturbed deterministic systems that are used as initial conditions for the stochastic simulations. The eventual source of problems is relate to the initial conditions for free growth factor. As both the spatially homogeneous steady state (0.47), as the end value of the perturbed system (< 0.2) are below 0.5, initial values for growth factor are always rounded to zero.

To verify if the initial values of free growth factor affect in a sensible way the evolution of the stochastic system, we modified two reaction propensities in order to rescale growth factor by a given factor so that the new level of growth factor is less sensitive to the rounding effect, without affecting the original level of pre-cancerous cells. To achieve this

- Equation 8.12 was replaced by

$$a_1 \frac{bc}{b + c \cdot resc_{GF}} \tag{8.30}$$

- Equation 8.19 was replaced by

$$\kappa \cdot resc_{GF} \cdot c(1 + c) \tag{8.31}$$

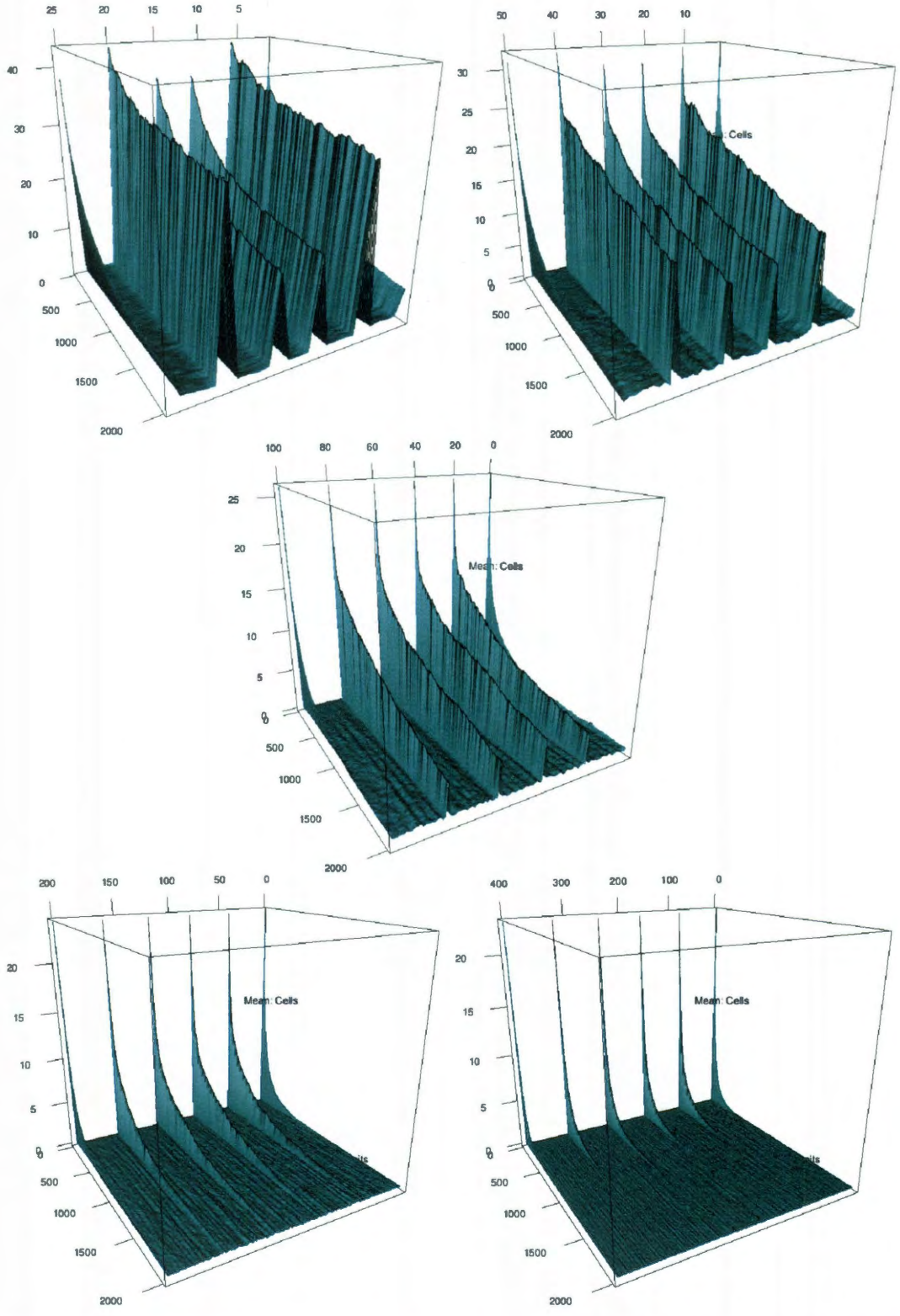


Figure 8.10 : Stochastic reaction-diffusion system. Averages of 1000 runs for different grid densities. The number of grid nodes appear on top of each panel.

were $resc_{GF}$ is the rescaling factor for growth factor chosen. It was selected $resc_{GF} = 100$, and the results for deterministic as well as stochastic systems are shown in Figure 8.11.

The perturbed deterministic system with growth factor rescaled looks exactly the same as the original (already presented in Figure 8.4), with the levels of free growth factor multiplied by 100. Interestingly, the same congruency seems to happen on the stochastic case. Pre-cancerous cells look almost indistinguishable, both in average as in standard deviation values, which seems to disregard a potential negative effect of the initial rounding to zero of free growth factor.

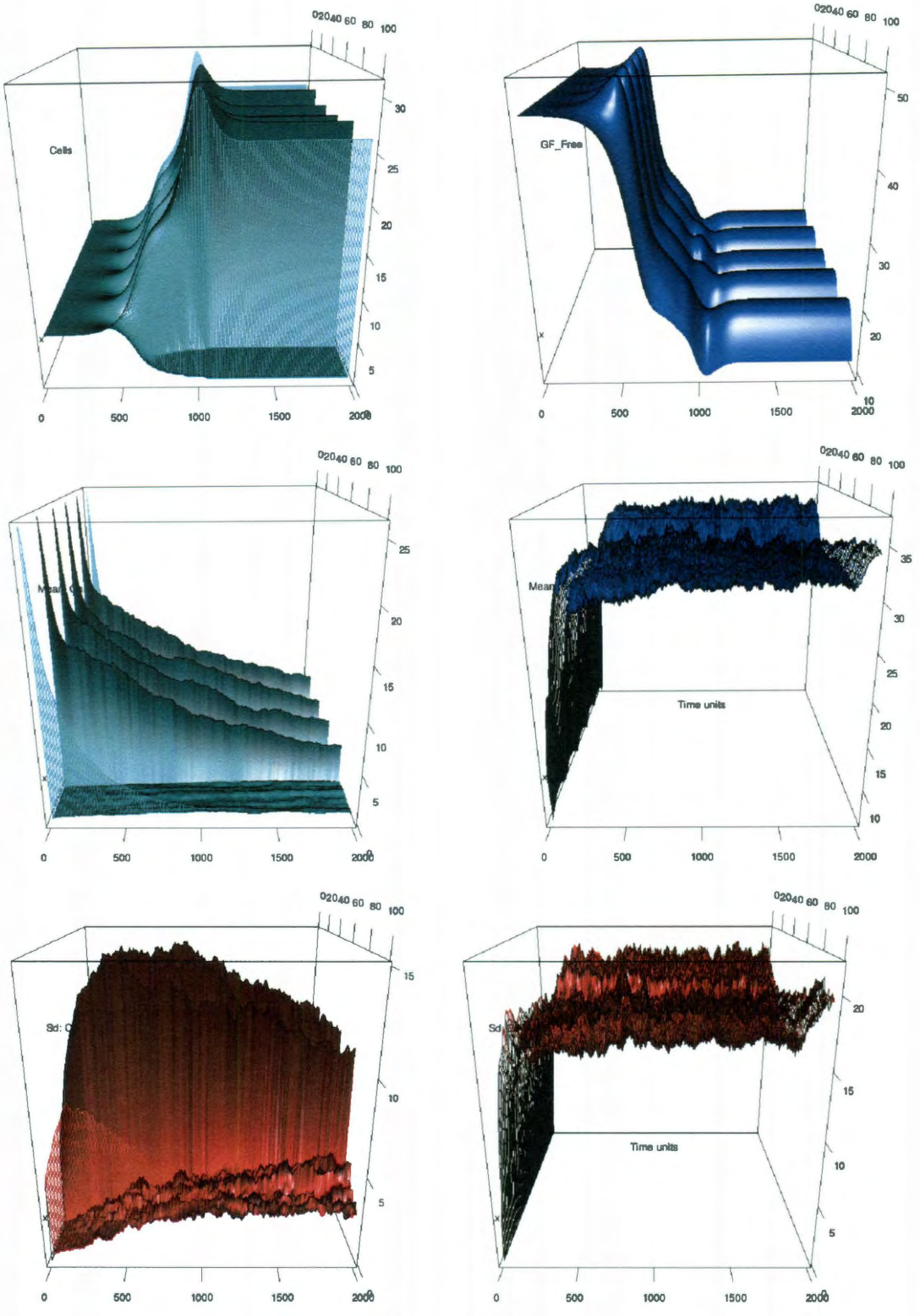


Figure 8.11 : Effect of rescaling growth factor in perturbed system. On the left are depicted pre-cancerous cells, while in the right are depicted free growth factor molecules. First row corresponds to deterministic evolution after perturbation. Second and third row correspond to averages and standard deviation, respectively, of 1000 stochastic runs with initial conditions set to the end values of the deterministic system rounded to nearest integer.

Chapter 9

Deterministic and stochastic reaction-diffusion model of viral infection with interferon production

In this chapter we compare deterministic and stochastic reaction-diffusion approaches in a two-dimensional model of viral infection with interferon production. It consists of five parts. In part one we explain the motivation for the study. In part two we introduce experimental conditions and how boundary conditions are treated. In part three we introduce the full model. In part four we construct the full model incrementally and we compare results of deterministic and stochastic systems at each step. In part five we detail the set of reactions of the full model.

9.1 Motivation

Haseltine et al. [26] published a deterministic simulation model, based on reaction-diffusion partial differential equations describing experiments involving in vitro interaction of viral spread and interferon defenses. Their predecessors [12] modeled the evolution of the infection front. We present calculations, based on Haseltine et al. [25, 26] data, which point at the need to use a stochastic model, and we build a model to compare deterministic and stochastic effects. Diffusion constants for viral particles and interferon are extracted from the published work of Haseltine et al. [26] group and their predecessors [12].

9.1.1 Need for a stochastic model

Haseltine et al. [26] built reaction-diffusion models to account for spatio-temporal patterns formed by the spreading viral infection, in systems with and without interferon-mediated antiviral response [12,25,26]. The experimental setup is shown in Figure 9.1.

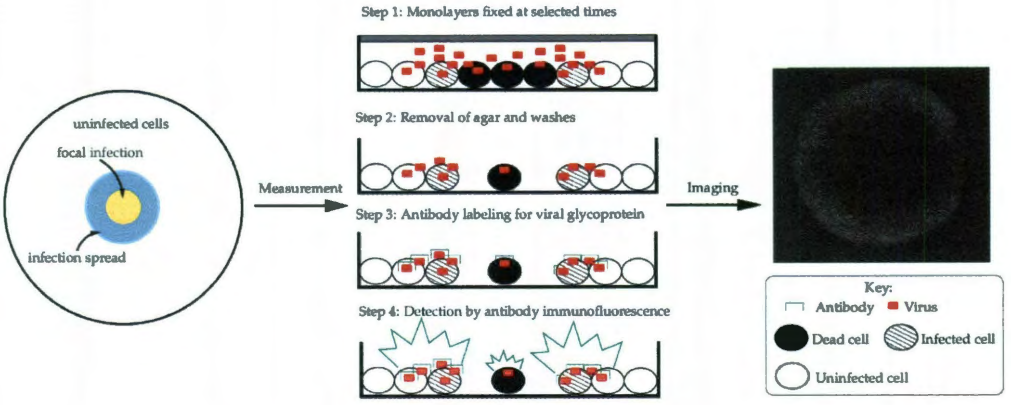


Figure 9.1 : Experiments of Haseltine et al. [25].

The model of Haseltine et al. is deterministic, although the authors comment that a possible extension might be exploring stochastic effects.

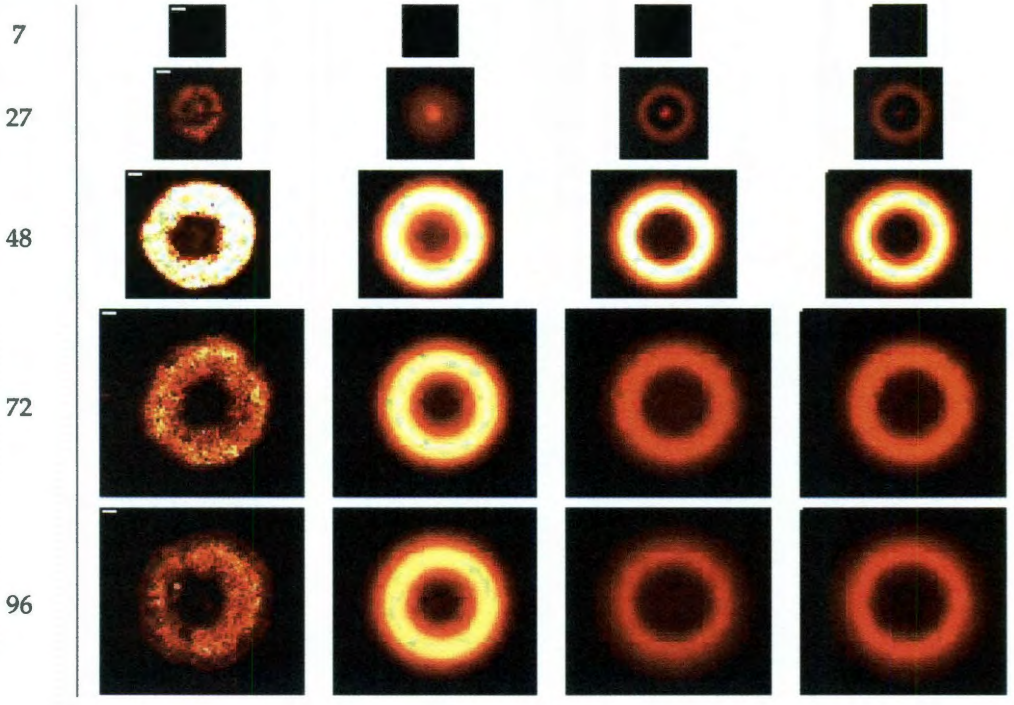


Figure 9.2 : Results of Haseltine et al. [25].

The experimental setup is as follows [25, 26]: In the center of a Petri dish of radius 1.75 cm a 1 mm radius circular opening is made using a pipette. We adopt all concentration and cell size measurements in Ref. [25,26]. Considering the cell volume

$V_c = 3.4 \times 10^{-9}$ ml, the radius of a spherical cell is:

$$r_c = \left(\frac{3V_c}{4\pi}\right)^{1/3} \cong 9.328 \cdot 10^{-6}\,\mathrm{m} = 9.328\,\mu\mathrm{m}. \tag{9.1}$$

The perimeter of the rim is $2\pi r$. If we accept that the boundary cells are located in the way shown in Figure 9.3, the approximate number of cells on the boundary is:

$$n_{\mathrm{cbound}} = \frac{2\pi r}{2r_c} \cong 337\,\mathrm{cells}. \tag{9.2}$$

In the opening $n_{\mathrm{vir},0} = 8.0 \cdot 10^4$ virions are inoculated. Assuming they are perfectly mixed, we only consider infection by virions that are at one cell radius distance from the rim (between the dashed-line circle and the solid-line circle in Figure 9.3).

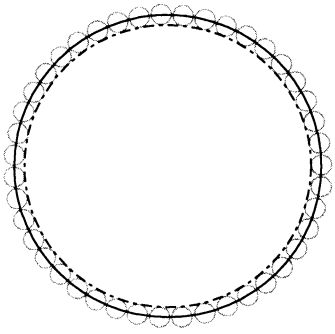


Figure 9.3 : Viral inoculum region with boundary cells. The outer circle ($r = 1$ mm) represents the rim of the hole made by the pipette. Gray circles represent cells on that boundary. The inner circle represents the limiting region where viruses are consider to be close enough to react with the cells. Not drawn to scale.

The number of virions within this ring-shape region is:

$$n_{\mathrm{virbound},0} = n_{\mathrm{vir},0}\frac{r_c(2r - r_c)}{r^2} \cong 1486\,\mathrm{virions} \tag{9.3}$$

Finally, the number of virions in the ring per cell is approximately equal to

$$n_{\mathrm{vir/cell}} = \frac{n_{\mathrm{vir},0}}{\pi}\frac{r_c^2}{r^3}(2r - r_c) \cong 4.41\,\mathrm{virions/cell} \tag{9.4}$$

Considering a Poisson distribution of the number of virions infecting a cell, the mean and variance are equal to $n_{vir/cell}$. The coefficient of variation is then equal to

$$\mathrm{cv} = \frac{\sqrt{n_{\mathrm{vir/cell}}}}{n_{\mathrm{vir/cell}}} \cong 0.48 \tag{9.5}$$

This simplified calculation indicates the importance of stochastic fluctuations in low-concentration viral in vitro and in vivo infection.

9.2 Experimental conditions

The background for the experimental conditions was extracted from Haseltine et al. [25, 26, 28].

Cells are grown at a density of 1×10^6 per 35 mm diameter well. Haseltine et al. assumed that the infection propagation is approximately radially symmetric, after noting that some of the experimental images demonstrated radial asymmetry. Therefore they modeled a one-dimensional reaction-diffusion system, discretizing the spatial dimension, using central differences, with an increment of 0.25 mm [25,26,28], that corresponds to 140 grid nodes.

We do not assume radial symmetry and study the two dimensional problem, inscribing the well in a square of 35 mm per side. Cell density of 1×10^6 per well corresponds to almost 65 cells per compartment. In order to have a square representation of cells for display purposes, we study $8 \times 8 = 64$ cells per compartment. As a result, we consider a total of 1,254,400 cells in a square of 35 mm per side on a grid of $140 \times 140 = 19,600$ compartments (or voxels in sbioPN terminology).

Haseltine et al. studied non age-dependent and age-dependent infection. In the latter case they discretized the age dimension using 17 points [26]. We discretize infection in 5 stages, that will be described later. Non age-dependent infection is considered as well in the last of the intermediate incremental models.

Our intention is to compare deterministic and stochastic systems in a model that describes in-vitro viral infection in a general way. We do not pretend to analyze any specific viral strain nor cell line. We use arbitrary rate constants beyond the diffusion rates. We reviewed a number of combinations of parameter values, and found that the systems are not highly sensitive to small variations from the assumed values, which are presented in Table 9.1. However, the parameter space is eleven-dimensional, so it is most likely that other combinations of parameter values might result in radically different dynamics.

9.2.0.1 Diffusion and boundary conditions

The rates of diffusion were extracted from Haseltine's PhD thesis [25]. As in the case of interferon two independent rates are provided, one by Porterfield et al. [57] and the other by Nichol and Deutsch [50], we use the mean value of both. The rate of diffusion of the vesicular stomatitis virus used is due to Ware et al. [68].

Only viral particles and interferon molecules diffuse. We choose periodic boundary conditions to have the same computational scheme for both deterministic and stochastic systems. Neumann boundary conditions, which might seem the natural choice, require two different computational schemes for the deterministic and stochastic cases, as it was explained in Chapter 7 (page 89), and applied in Chapter 8.

In our simulations, the infection is started in the middle of the grid, so boundary conditions only affect the results at late times.

The implementation of periodic boundary conditions is illustrated in Figure F.3 on page 213. The R code to generate the required pattern is listed in Section F.4 (page 215). Effective rates of diffusion were calculated using Equation 6.13 (page 80).

9.3 Description of the full model

A diagrammatic representation of the full model is shown in Figure 9.4.

The model includes the following objects:

- **UC**: Uninfected cell.
- **IC$_i$**: Infected cell, i-th stage ($i = 1, \cdots, 5$).
- **RC**: Resistant cell.
- **Vir**: Virion.
- **Ifn**: Interferon molecule.

Virions and interferon molecules diffuse, whereas cells remain in fixed grid compartments.

Uninfected cells (UC) bind competitively to viral particles (Vir), becoming infected (IC1), and to interferon molecules (Ifn), becoming resistant (RC).

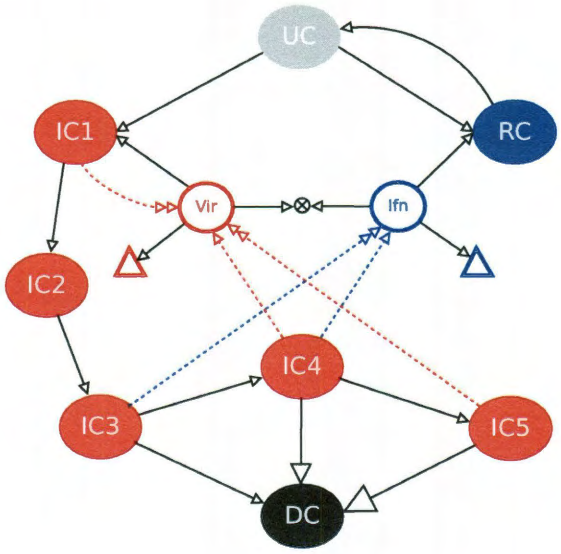


Figure 9.4 : Full model diagram.

We consider in the full model a discrete five-stage infection of the cell. First and second stages are dormant. In the first stage, viral particles uncoat releasing their genome and use the cell's machinery to produce viral RNA and proteins. In the second stage, innate immune system recognizes the presence of viral RNA and activates defensive and apoptotic pathways.

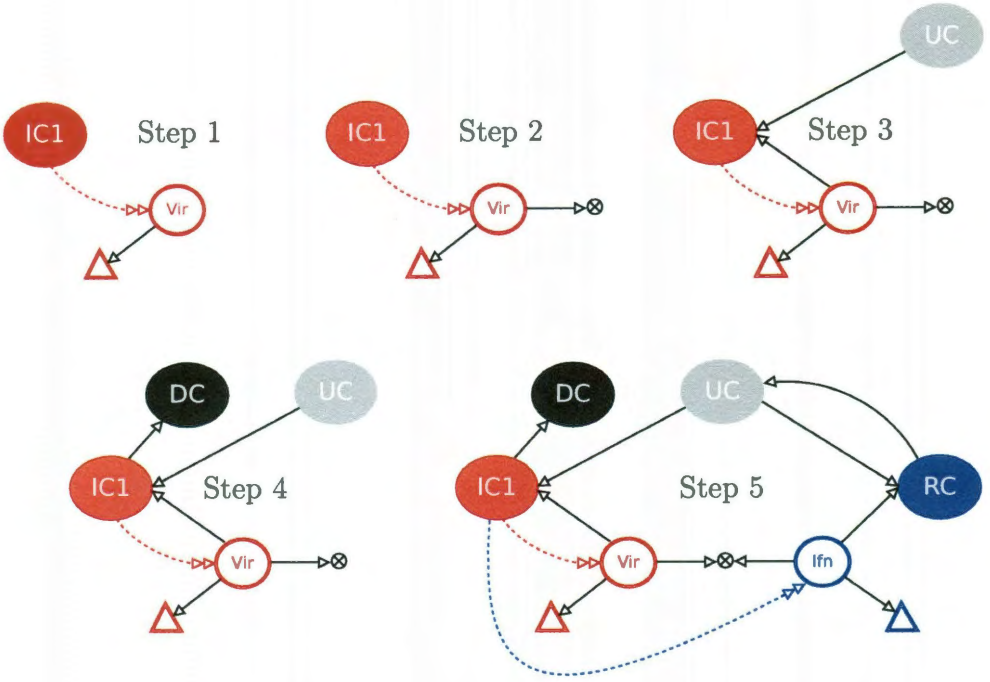


Figure 9.5 : Incremental construction of the model: steps 1 to 5.

In our model, cells at third stage of infection produce interferon at full rate, while cell at fourth stage of infection produce interferon and virions at half rate. Finally, cells at fifth stage of infection produce viral particles at full rate. Infected cells at stages 3, 4, and 5 have a chance to die (or enter apoptosis) at relative rates of 1/4, 1/2 and 1, respectively. Eventually, all infected cells die (DC).

As already mentioned, in the model only viral particles and interferon diffuse. Resistance in cells is temporary, and eventually resistant cells return to the pool of uninfected cells. To avoid adding extra second order reactions, once a cell is infected or resistant, it does not bind to further viral particles or interferon molecules. We compensate this by considering degradation of viral particles and interferon molecules.

9.4 Incremental construction of the model

To better understand when differences between deterministic and stochastic methods arise, we will construct the model in six incremental steps and analyze the results of

the incremental models. The rate constants, identical for each step, are summarized in Table 9.1.

The incremental steps include:

1. Viral production and diffusion of viral particles.
2. Viral particles internalization by cells, or viral particles degradation.
3. Cell infection.
4. Infected cell death.
5. Interferon production and acquisition of cell resistance.
6. Infection stages.

Diagrammatic representations for steps 1 to 5 are depicted in Figure 9.5, and for step 6, corresponding to the full model, in Figure 9.4.

9.4.1 Viral production and diffusion of viral particles

The simple initial model is composed by two species only: Infected cells (IC1) and viral particles (Vir). The corresponding coupled chemical equations are:

$$\mathrm{IC_1} \rightarrow \mathrm{IC_1} + \mathrm{Vir} \tag{9.6}$$

$$\mathrm{Vir} \rightarrow \Delta_{xy} \tag{9.7}$$

In this model we will assume as initial conditions 48 infected cells (3/4 of the total) in one of the four central compartments (as the grid has an even number of compartment and no unique central compartment exists). Virus is produced only in the central compartment, from where it diffuses.

The results of the deterministic and stochastic runs are presented in the top portion of Figure 9.6. Deterministic and stochastic systems closely agree. We show a

close-up (as intensity of viral diffusion is low) of the state of viral particles for both systems at 100 h.

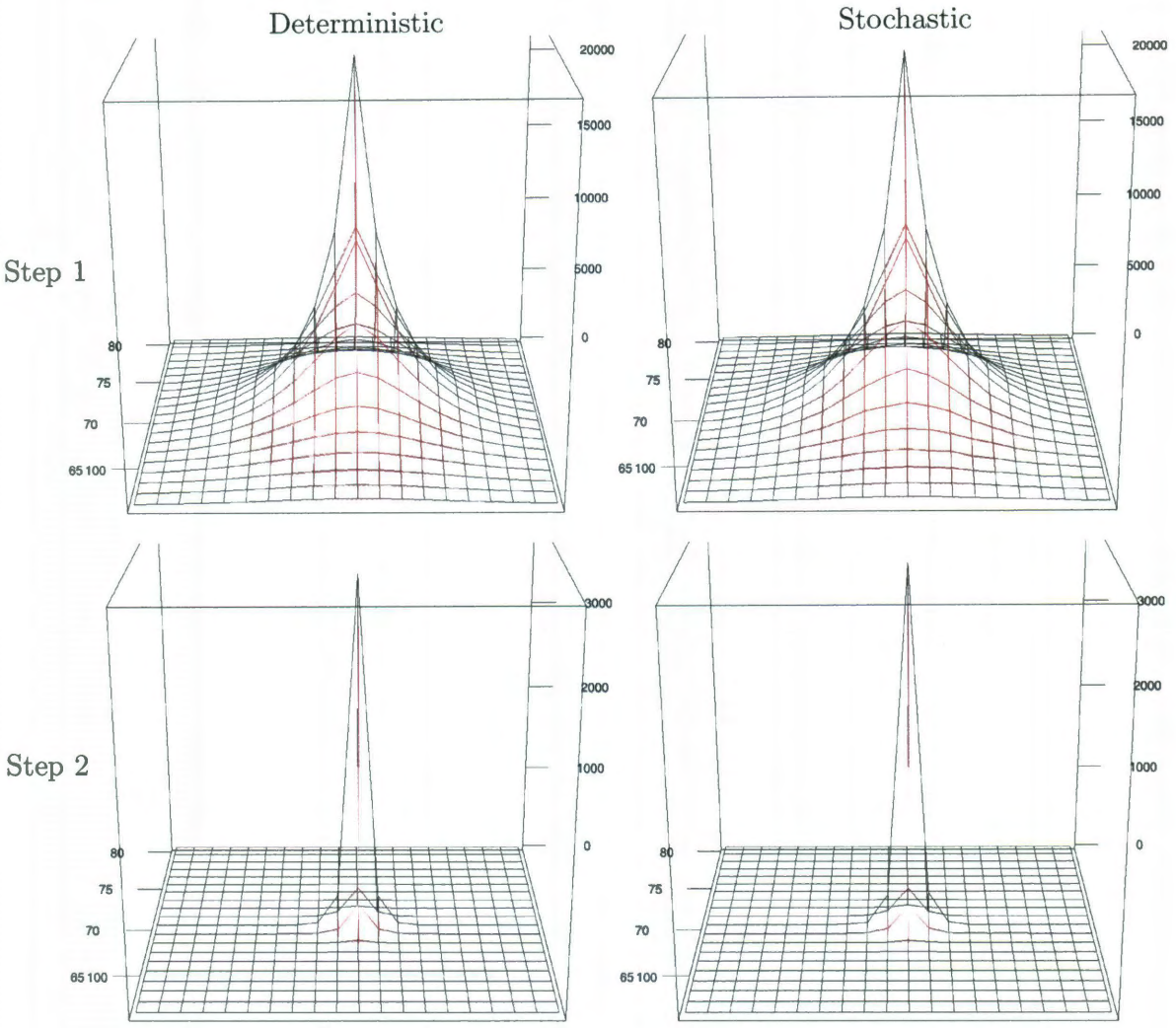


Figure 9.6 : Steps 1 and 2.

In Figure 9.7, we present a view from above that shows that evolution of both deterministic and stochastic systems is radially symmetric with center in the central compartment and of similar intensity.

9.4.2 Viral particles internalized by cells, or viral particles degradation

In this section, we model internalization of viral particles that do not change the status of the cell. As already mentioned, we assume that once a cell is infected, new

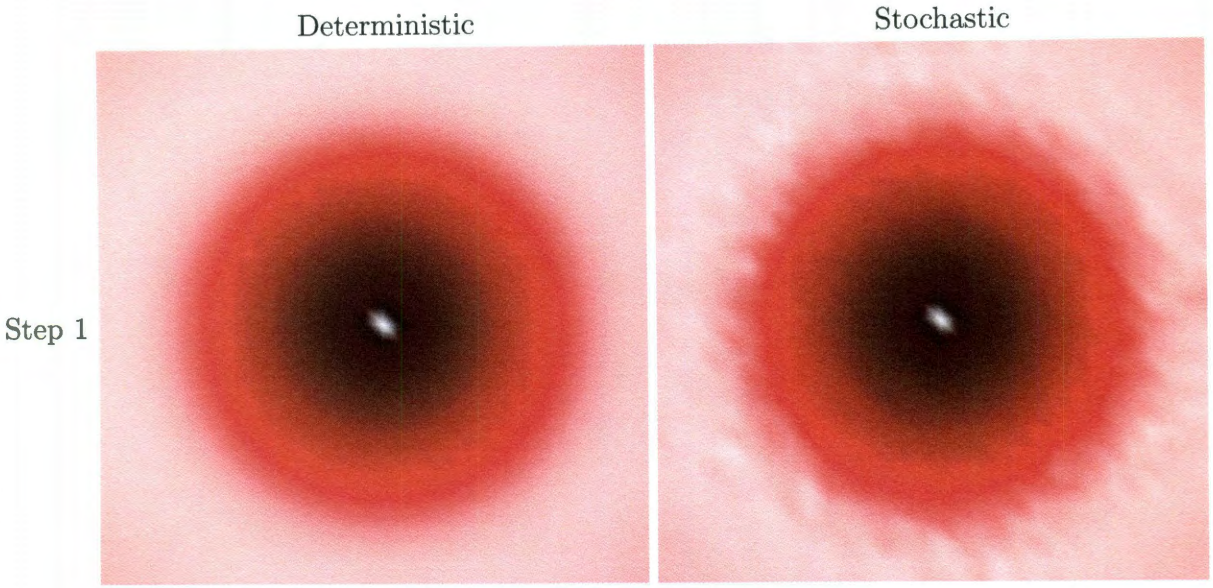


Figure 9.7 : Step 1. View from above.

internalized viral particles do not increase viral particles yield. Moreover, we assume that not all viral particles internalized by an uninfected cell infect it. We assume the first order reaction of viral particles degradation.

The corresponding new reaction is:

$$\text{Vir} \longrightarrow \otimes \tag{9.8}$$

Compared to the model of the previous step, the distance that a viral particle can reach from the central compartment is diminished considerably, even bounded in the deterministic case. We show a close-up of results at 100 hours (lower panel of Figure 9.6). It is important to note that the evolution looks unaltered since the first few hours until the end of runs. Both deterministic and stochastic runs continue to closely agree.

9.4.3 Viral infection

We model infection as a second order reaction: one viral particle (Vir) reacts with one uninfected cell (UC) and the product is an infected cell (IC1). Initial conditions, from now on, will be the following: at time zero, there are 64 uninfected cells in each compartment, with the exception of the central compartment in which there are six infected cells and 58 uninfected cells. All other quantities are set to zero.

This step adds the following reaction:

$$\mathrm{Vir} + \mathrm{UC} \longrightarrow \mathrm{IC}_1 \tag{9.9}$$

Results for viral particles are displayed in the top portion of Figure 9.8. Cells (uninfected in gray and infected in red), are displayed in the top portion of Figure 9.9.

The introduction of the second order reaction produces an important change in behavior of the system. In comparison with simpler previous models without infection, three things can be noted:

1. Both deterministic and stochastic infection models evolve faster than models without infection. Figure 9.6 shows close-ups at 100 hours. Figures 9.8 and 9.9 show views of the complete system at 5 hours, where the deterministic system is about to reach the boundaries of the squared well.

2. Radial symmetry is lost. Figure 9.7 shows a radially symmetric viral expansion, while in Figures 9.8 and 9.9 both the deterministic and stochastic systems exhibit a diamond shape. This is further analyzed in Section 9.6.

3. The deterministic system evolves faster than the stochastic system.

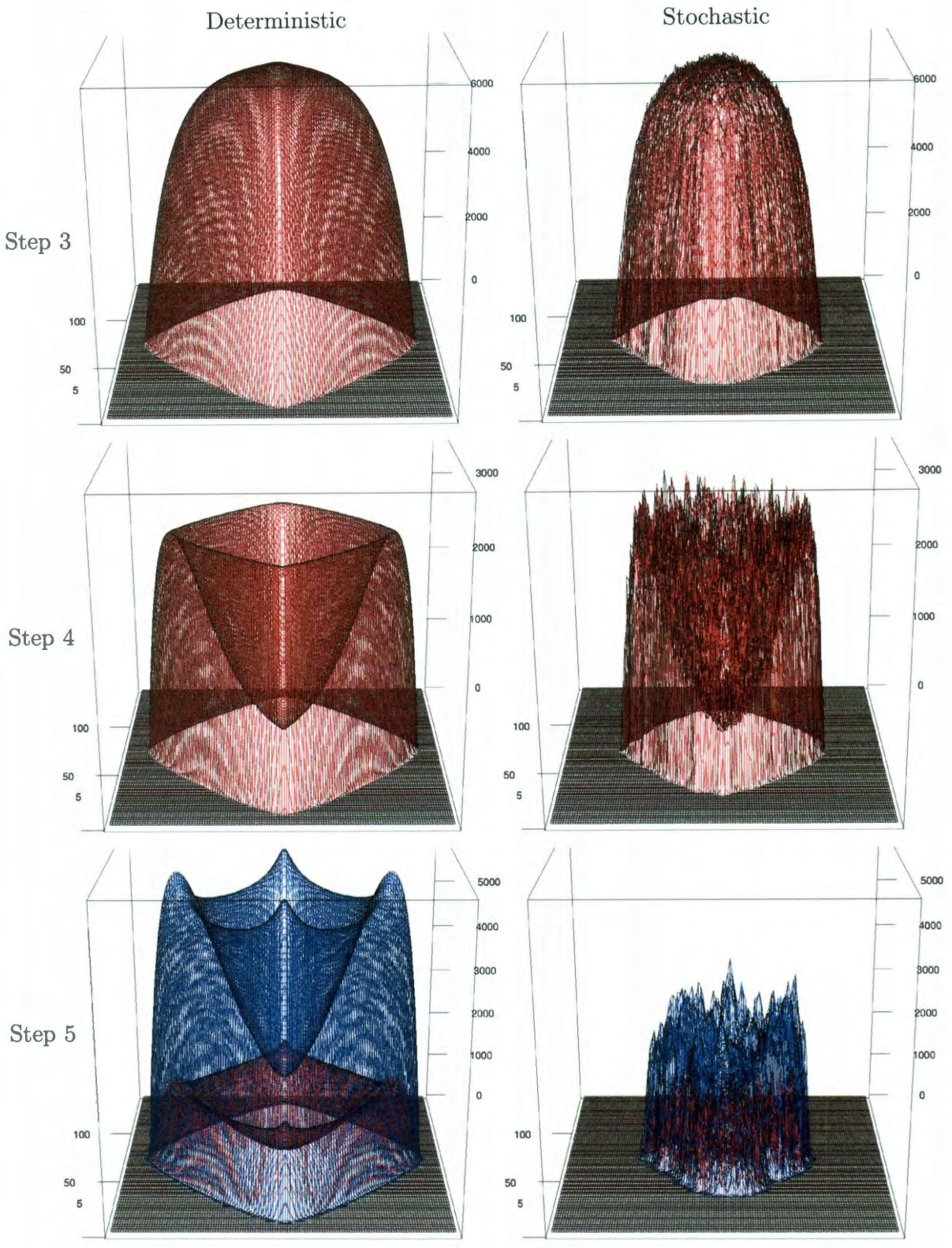


Figure 9.8 : Results for viral particles (in red) and interferon molecules (in blue), for steps 3 to 5, at 5 hours.

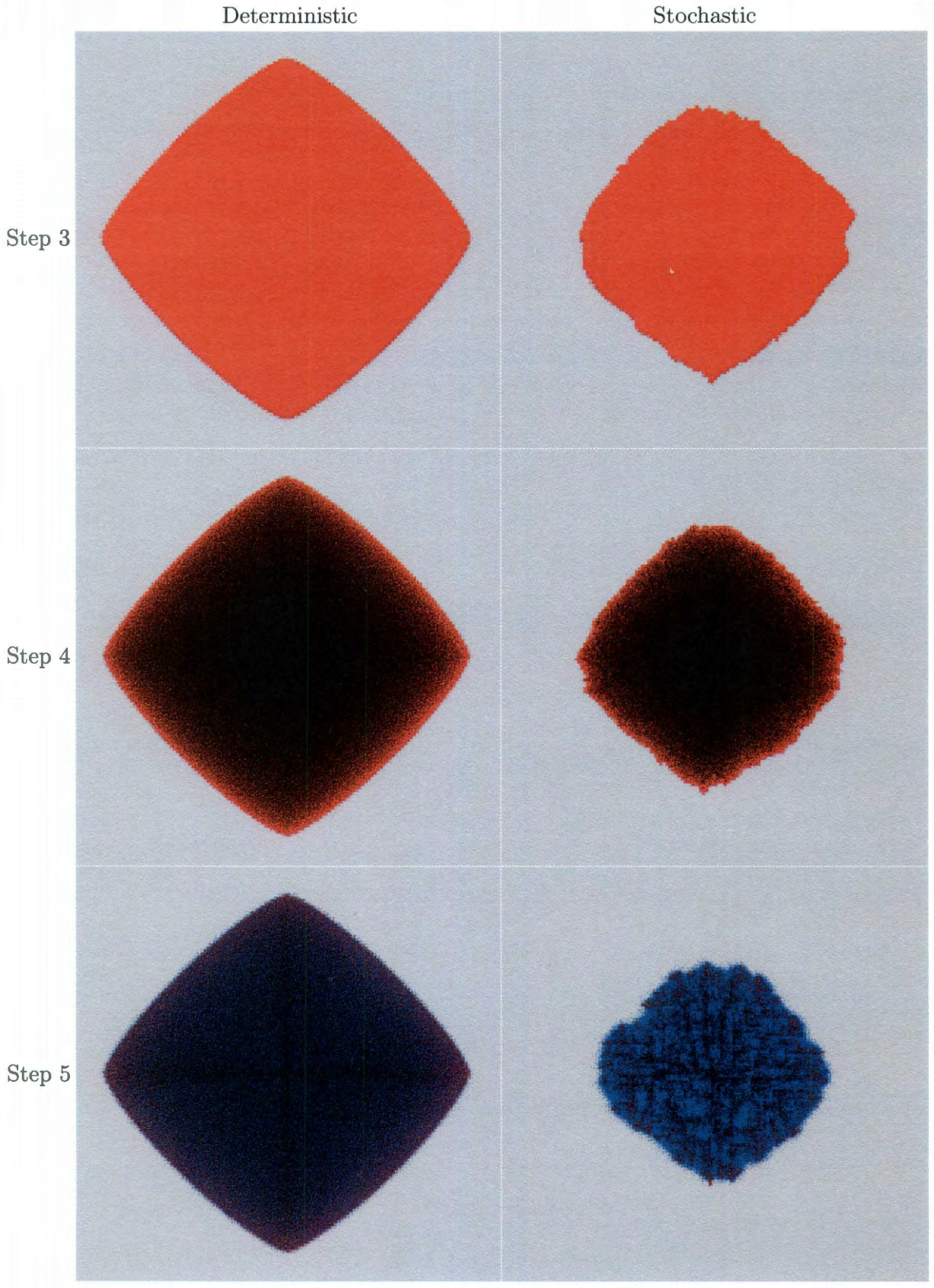


Figure 9.9 : Results for cells, for steps 3 to 5, at 5 hours.

9.4.4 Infected cell death

Death of an infected cell is modeled with the first order reaction:

$$\mathrm{IC}_1 \longrightarrow \mathrm{DC} \tag{9.10}$$

Results at 5 hours for viral particles are displayed in the middle portion of Figure 9.8. Cells (uninfected in gray, infected in red, and dead in black), are displayed in the middle portion of Figure 9.9.

Infection, in this case, seems to evolve with the same speed similar to the observed on the step 3 model. As cells die, they stop producing viral particles, which is observed in the convex shape the viral particle surface assumes. As in the previous model, the deterministic evolution is faster than the stochastic evolution, and the diamond-shape artifact is maintained.

9.4.5 Interferon production and cell resistance

Infected cell produce interferon (Ifn), which binds to uninfected cells making them resistant (RC). This last reaction is of second order. Interferon binds to cells, no matter which the cell status is. However, to avoid including second order reactions, we follow the same approach as done with viral particles, and assume that interferon degrades.

This step adds the following reactions:

$$\mathrm{IC}_1 \longrightarrow \mathrm{IC}_1 + \mathrm{Ifn} \tag{9.11}$$

$$\mathrm{Ifn} + \mathrm{UC} \longrightarrow \mathrm{RC} \tag{9.12}$$

$$\mathrm{RC} \longrightarrow \mathrm{UC} \tag{9.13}$$

$$\mathrm{Ifn} \longrightarrow \otimes \tag{9.14}$$

$$\mathrm{Ifn} \longrightarrow \Delta_{xy} \tag{9.15}$$

Results at 5 hours are depicted in the bottom portions of Figures 9.8 and 9.9. Interferon particles and resistant cells are displayed in blue. The deterministic system continues to evolve faster than the stochastic system and it does not capture the structure that the stochastic system exhibits.

9.5 Full model, with infection stages

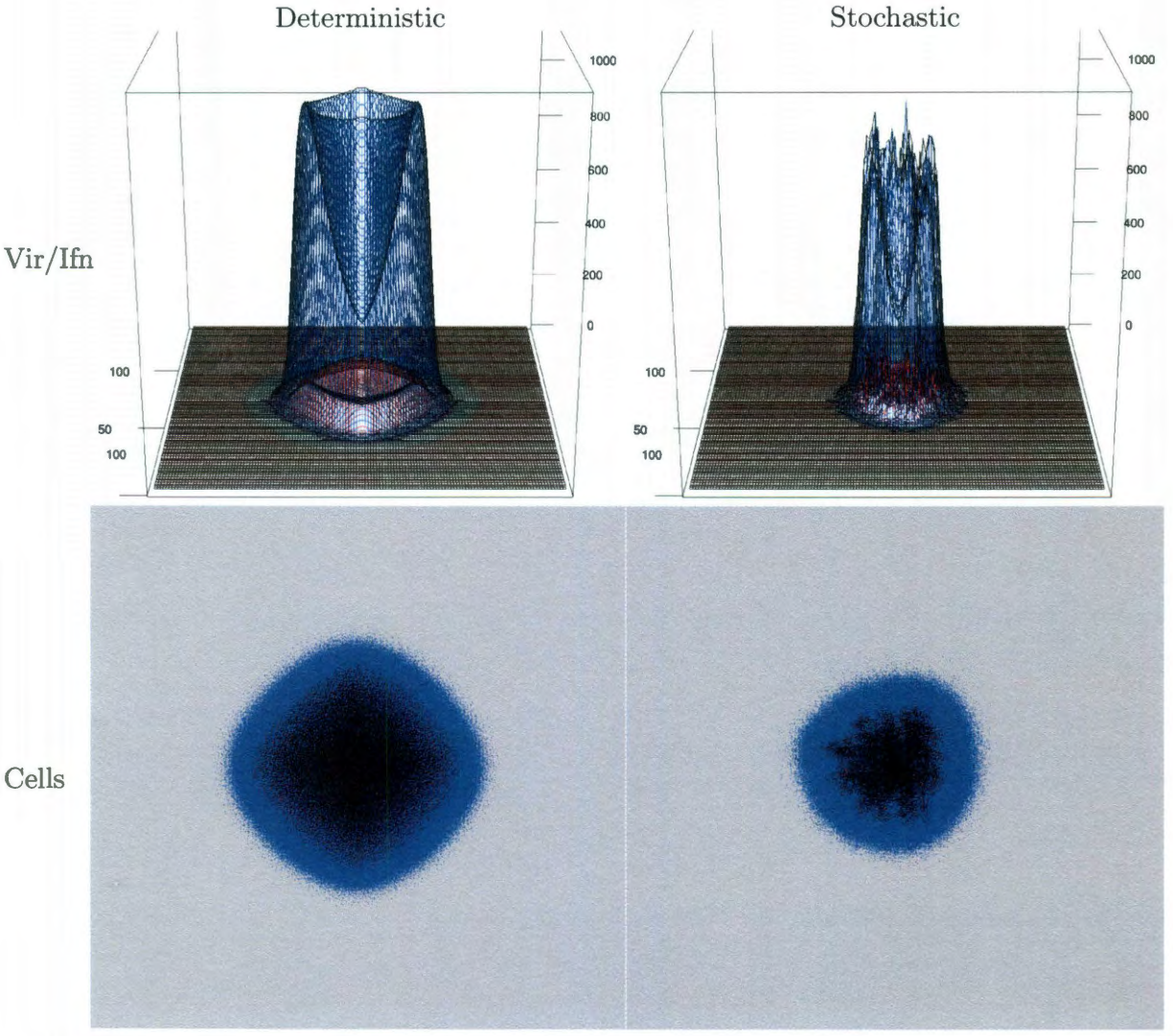


Figure 9.10 : Results for full model at 100 hours.

The final incremental step adds five infection stages, which were already described. The set of coupled chemical reactions with the corresponding explanation, are pre-

sented in Section 9.9. The diagrammatic representation is depicted in Figure 9.4.

Results at 100 hours for the full model are presented in Figure 9.10. In contrast with models without infection age, this model evolves orders of magnitude slower. As in previous cases since infection was introduced, the deterministic system evolves faster than the stochastic counterpart, and the stochastic system exhibits structure that is not captured by the deterministic system. The diamond-shape artifact is still present, but the effect is less noticeable in the external front of the resistant cells than in the internal front of the infected and dead cells.

9.6 Analysis of diamond-shape artifact and remedial measures

As the diamond-shape artifact first appears in the step 3 model, we analyze this simpler case as it only has three reactions (viral production, viral particle degradation, and cell infection) and one diffusion element (viral particles).

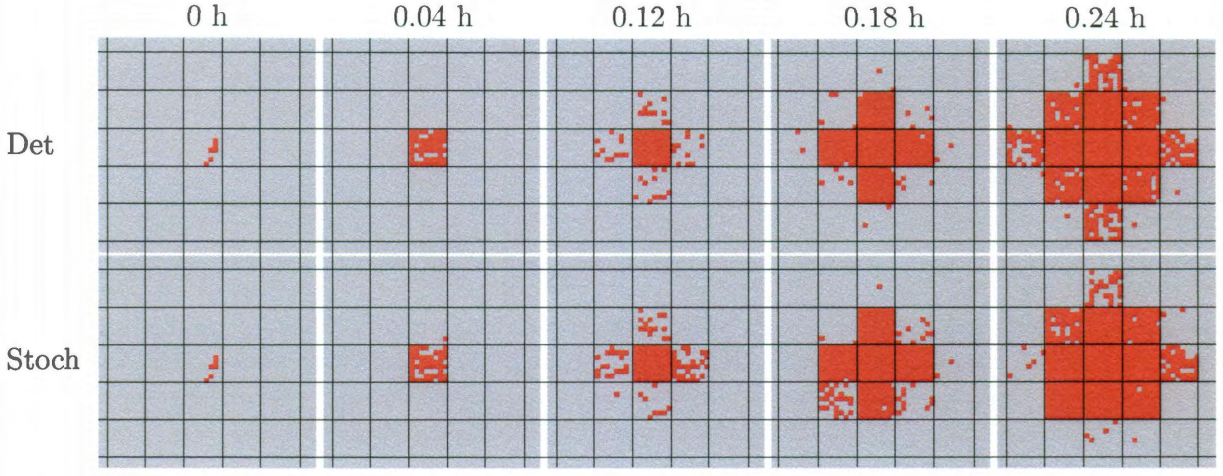


Figure 9.11 : Close-up view of early times of viral infection, corresponding to the model with viral infection added. Squares represent compartments, each populated by 64 cells (infected cells in red, uninfected cells in gray). The complete system is partitioned in 140x140 compartments.

Figure 9.11 shows close-ups of evolution of uninfected (gray) and infected cells (red) at early time points (0, 0.04, 0.12, 0.18, and 0.24 hours). The deterministic

run is shown at the top and the stochastic run at the bottom. There are 64 cells per compartment, and cells are displayed in 8×8 squares.

The left column shows the initial state: 6 infected cell (in red) and 58 uninfected cells (in gray) on one of the four central compartments. The rest are uninfected cells.

The second column (0.04 hours) shows that most cells in the central compartment are infected, but only one cell, and only for the stochastic run, is infected in a different compartment.

The third column (0.12 hours) shows that all the cells of the central compartment are infected, as well as about one third of the cells in each of the compartments that share a side with the central compartment. Only one cell is infected, again for the stochastic run, on one of the compartments that only share a vertex with the central compartment.

The fourth column (0.18 hours) shows a cross shape where all the side compartments are almost full of infected cells.

The last column (0.24 hours) shows a square of 3×3 compartments almost full of infected cells, and new arms of infection evolving in a preferred vertical and horizontal directions, that will evolve in a diamond shape. In this last column it is evident that the deterministic system starts evolving faster than the stochastic one, while in the previous two columns, at least for this stochastic run, the impression is the contrary.

This artifact is related to the fact that diffusion rate is too small when compared with the size of the compartment and the infection rate. As well mixed assumptions inside compartment are made, once infection starts in a given compartment, the new generated viral particles are more likely to infect local cells than to diffuse to neighbor compartments. Local infected cells produce more viral particles. As the number of viral particles increase, so does the propensity associated with the diffusion term, and viral particles eventually reach neighbor compartments and a new local infection starts. It is then the anisotropy of the square grid system that favors a vertical and

horizontal expansion what produces the diamond shape observed.

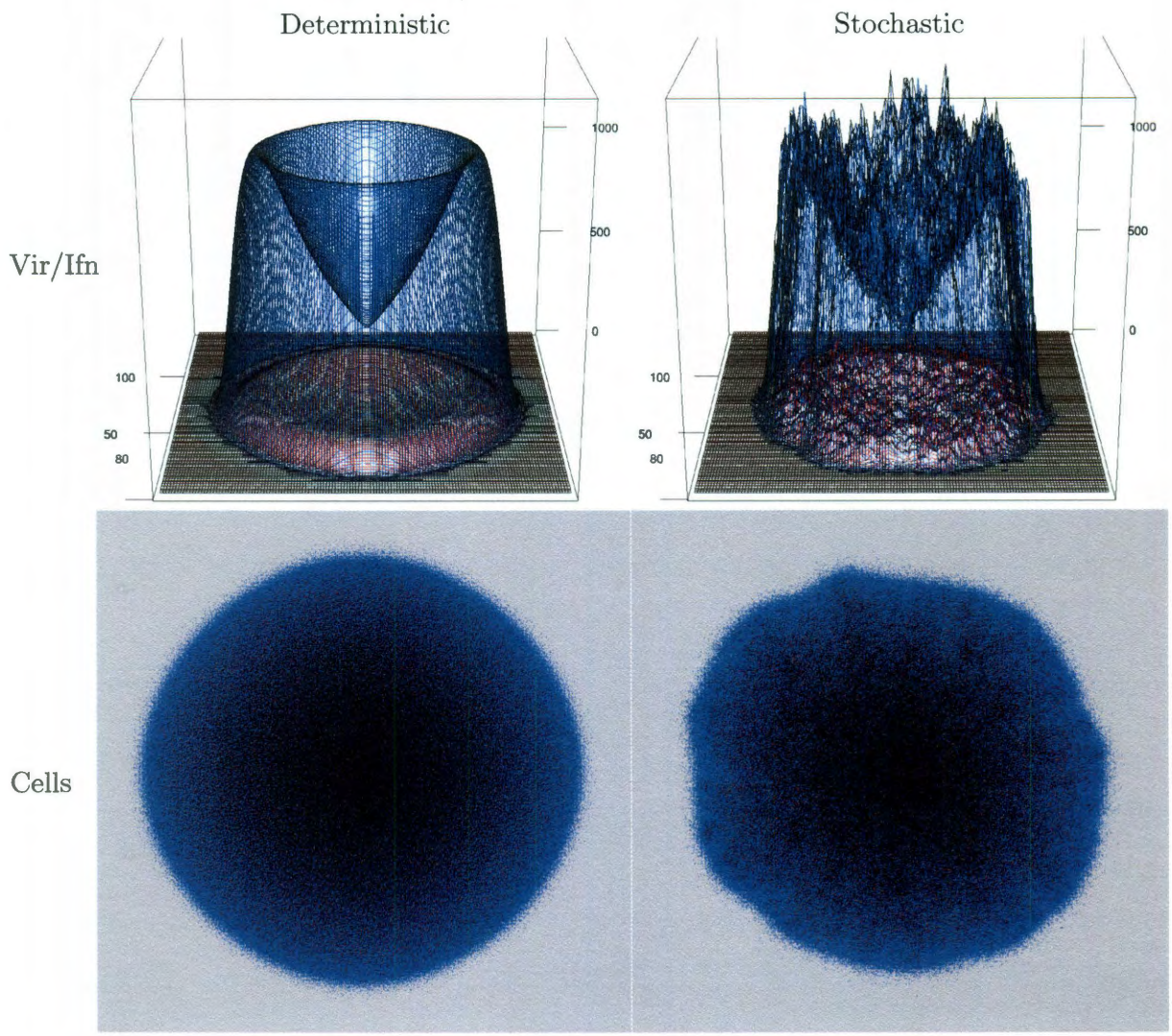


Figure 9.12 : Full model with viral diffusion rate increased by an order of magnitude.

If we increase the rate of diffusion by a corresponding factor, we observe an infection evolving with a radially symmetric shape. We obtained almost perfect circles for the deterministic case by increasing the diffusion rate by two orders of magnitude. Results for the complete model where a perfectly radially symmetric deterministic infection expansion is obtained by increasing the diffusion rate by one order of magnitude are shown in Figure 9.12. The lower needed increase of diffusion for the full model seems to be related to the fact that infection progresses orders of magnitude

slower than in the step 3 model, and this allows viral particles to diffuse faster than the infection front, conveying the expected radially symmetric diffusive shape, instead of the infective diamond shape. The deterministic system still evolves faster than the stochastic system, but there is less discrepancy observed.

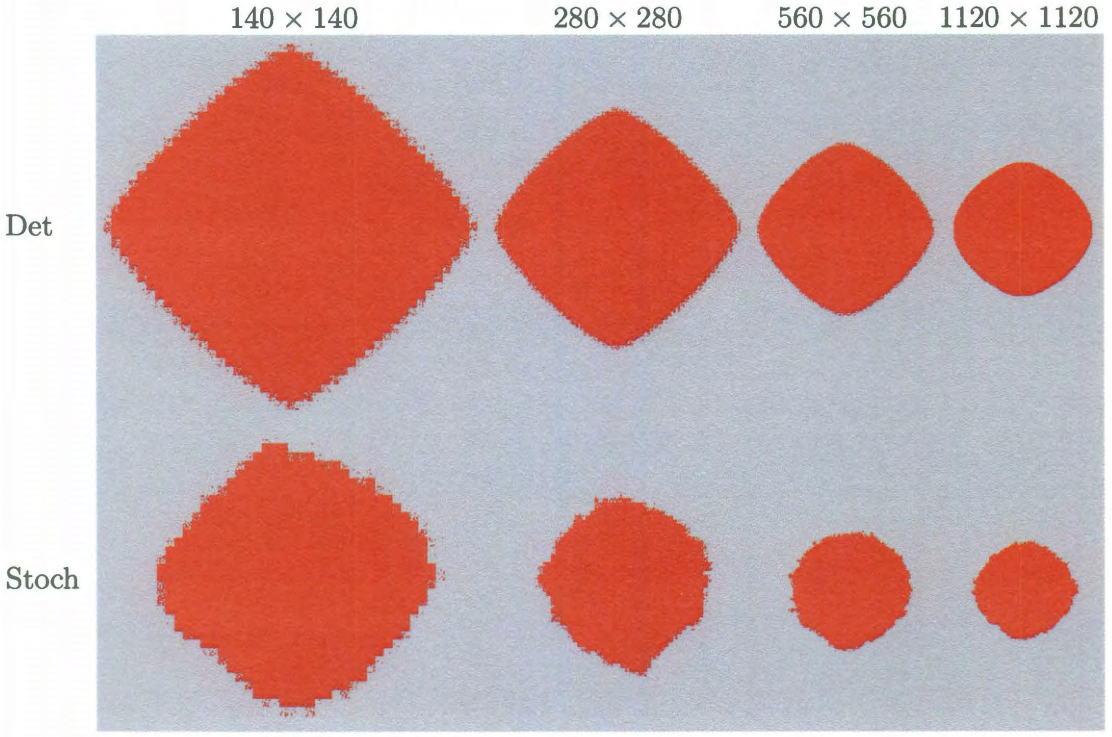


Figure 9.13 : Comparison of deterministic and stochastic infection-only model for different compartment sizes, step 3 model. All cases display infected cells (red) at 1.8 hours. The same scale applies to all figures. Legends on top denotes number of compartments. From left to right, each compartment has: 64, 16, 4, and 1 cells respectively. Each case has its compartment area equal to one quarter the compartment area of the case on its left.

The effect of reducing the compartment area by increasing the number of compartments is shown in Figure 9.13. As the number of compartment increases (or the area of each compartment decreases):

- the diamond-shape artifact in both the deterministic and the stochastic systems becomes less noticeable.
- the infection spread speeds in both the deterministic and the stochastic systems slow down.

- the comparative infection spread of the deterministic and the stochastic systems does not change: the deterministic system evolves faster than its corresponding stochastic system.

Figure 9.14 shows the same corrective effect happening in the full model. In this case the diamond-shape artifact completely disappears for a partition of the whole system in 560 × 560 compartments. It can also be observed here than the infection spread speed also decreases as the number of compartments increase.

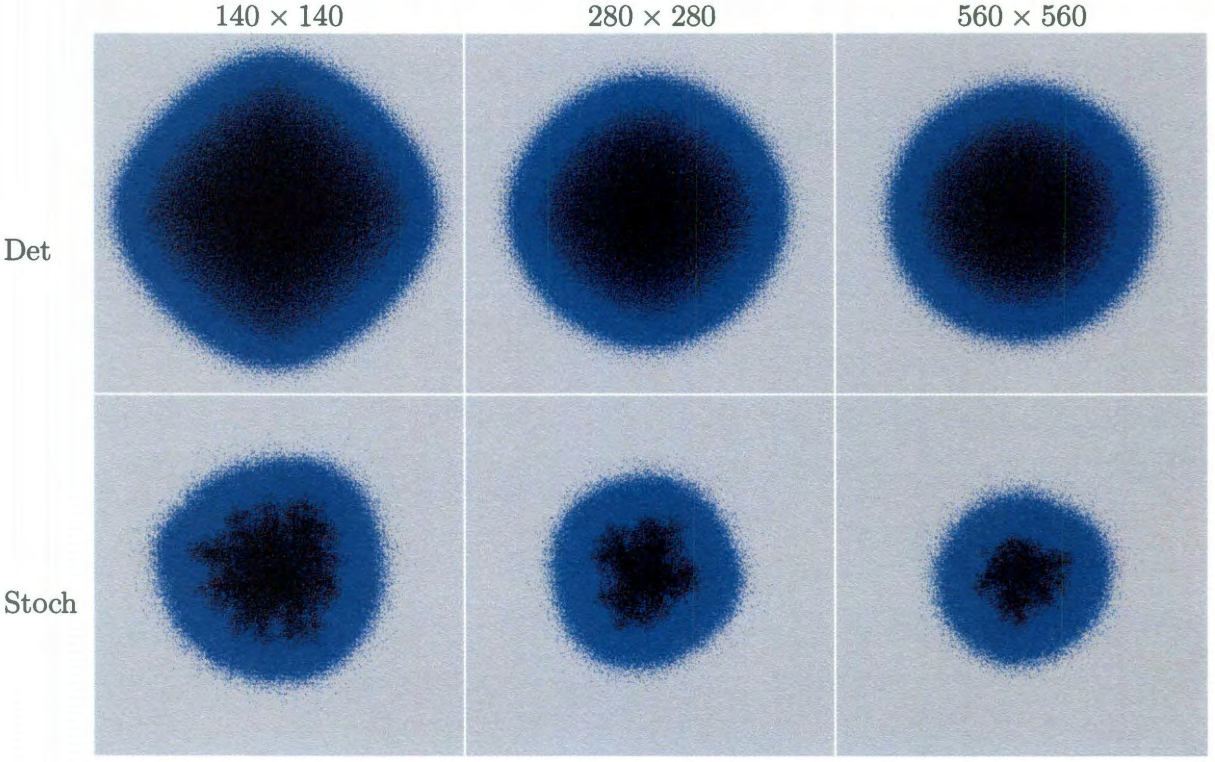


Figure 9.14 : Results for the full model for different grid densities.

The diamond-shape artifact seems to play a diagnostic role, as it disappears when a balanced selection of compartment size and of diffusion, infection, viral production and degradation rate constants are chosen. The use of a different compartment shape, such as triangular or hexagonal, would certainly reduce the diamond-shape artifact. However, the associated problem of infection spread speed may not be detected in those cases.

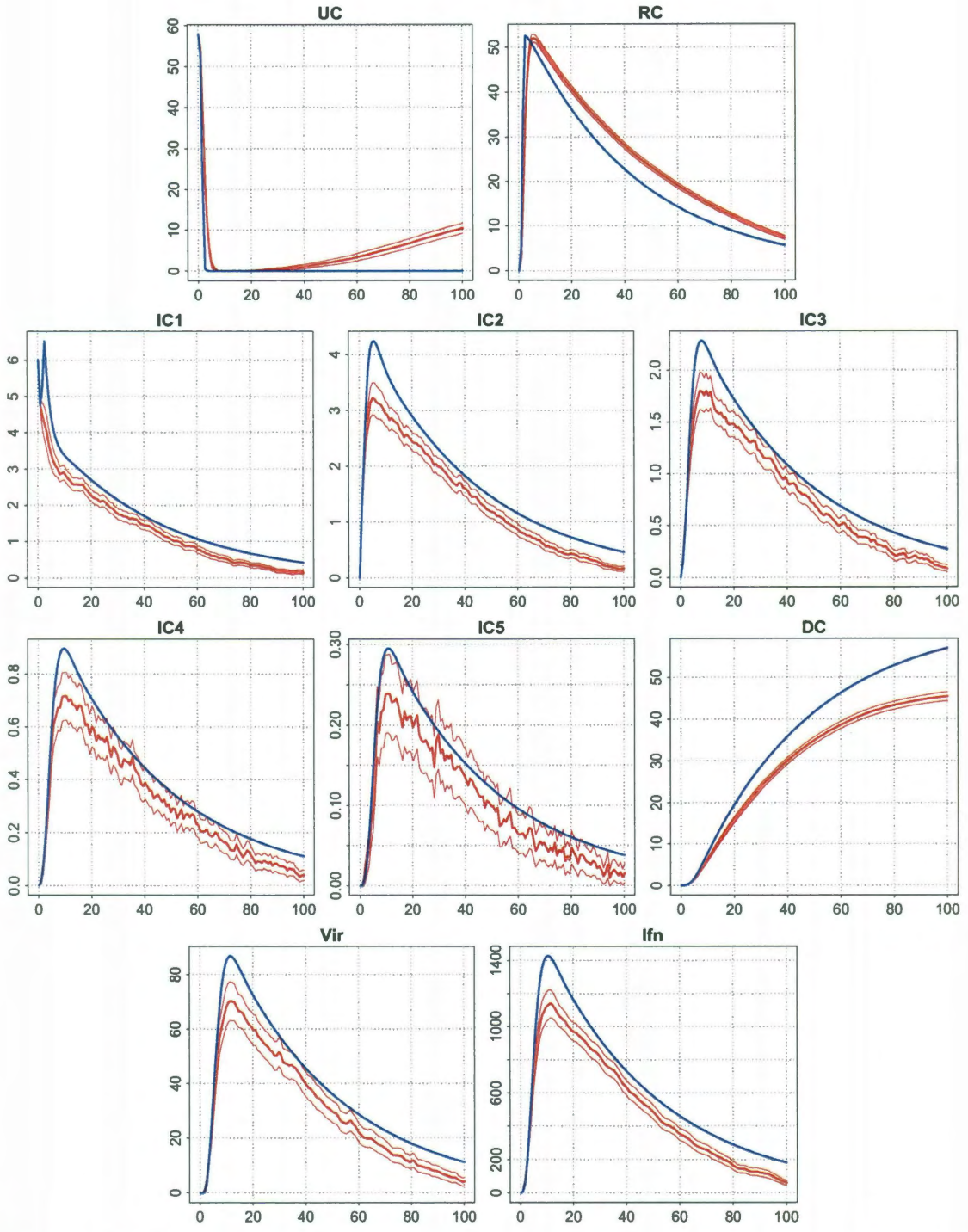


Figure 9.15 : Deterministic and stochastic comparison for the non-spatial full model. Vertical axes: number of units. Horizontal axes: time in hours. Deterministic evolution is depicted in blue. Stochastic mean $\pm$ 3 standard errors (99.73% confidence band) is depicted in red.

9.7 Analysis of difference in evolution speed between deterministic and stochastic simulations: non spatial model

The model without diffusion is shown in Figure 9.15. Blue lines depict deterministic runs, red lines depict averages of 1000 runs, and dotted red lines depict standard deviations of 1000 runs. In line with the reaction-diffusive systems, deterministic evolutions over-estimate or under-estimate the mean of the stochastic process.

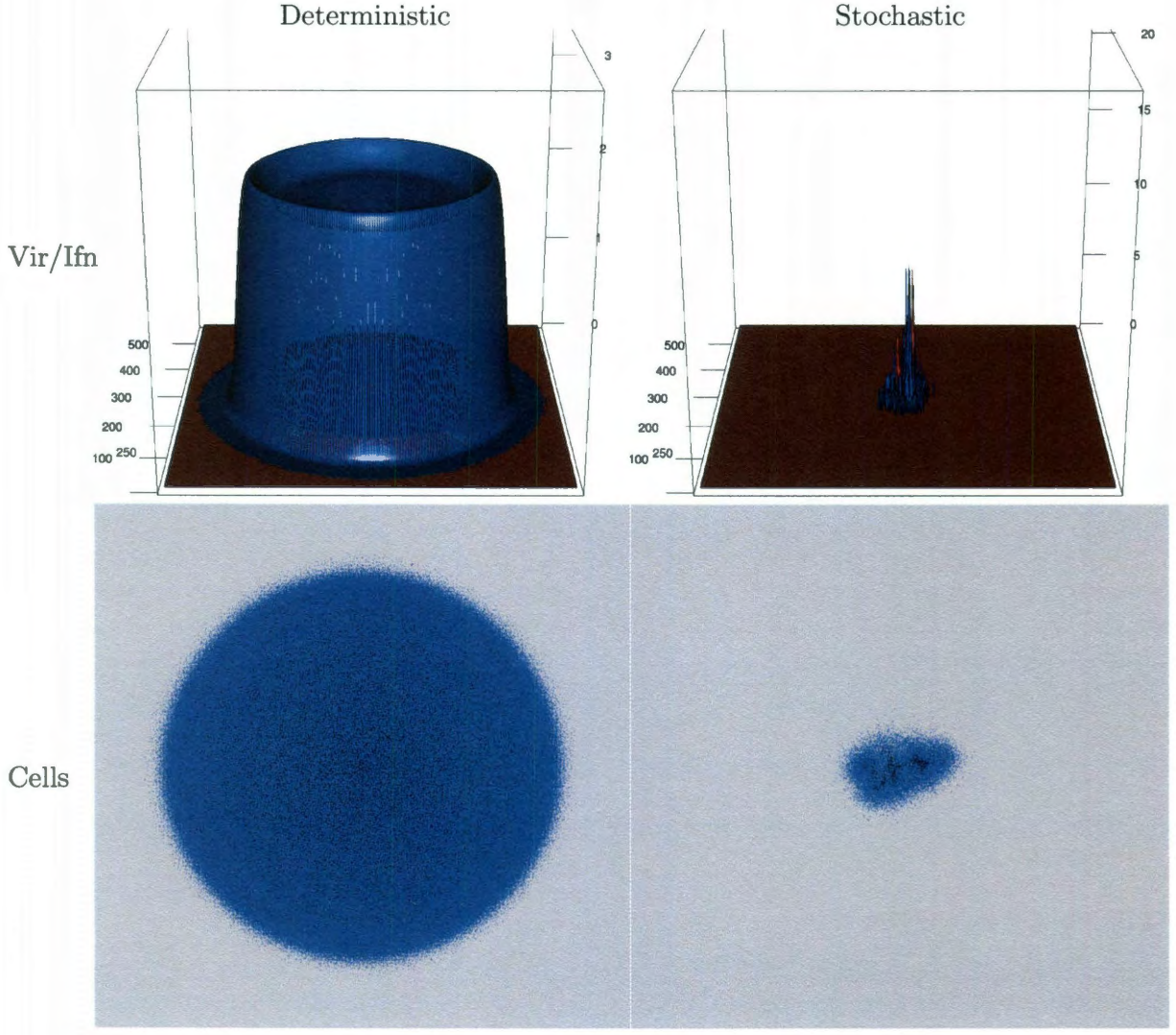


Figure 9.16 : Full model with infection rate = $1/(250 \text{ mol h})$, at 250 hours, for the system partitioned in 560×560 compartments.

As the non-spatial system shows discrepancy between the deterministic and stochas-

tic formulations, due to the fact that some reactions are of second order, the associated diffusive models will also exhibit the same behavior.

9.8 Effect of decreasing infection rate

Figure 9.16 shows the effect of decreasing infection rate to 1/(250 mol h). The deterministic system completely fails to capture the evolution of the stochastic system: while infection in the stochastic system performs a sort of random walk around the central region for extended simulation times (Figure 9.17), that includes splitting and rejoining of small infection fronts, the deterministic evolves radially, as in previous cases.

9.9 Set of chemical reactions of the full model

All the reactions considered follow mass action kinetic laws. For further clarification, see 2.3.1 on page 11.

9.9.0.1 First stage of infection: viral particle gets internalized by uninfected cell

Viral particles interact with uninfected cells in a competitive fashion with interferon molecules (see Equation 9.28).

$$\mathrm{Vir} + \mathrm{UC} \xrightarrow{\text{UC\_infection\_r / vir\_avg}} \mathrm{IC}_1 \tag{9.16}$$

The viral particle is consumed in the process. We do not consider ulterior interactions between viral particles and cells at other stages other than uninfected. Viral particles are eliminated from the system according to Equation 9.31.

In this first stage of infection, the capsid of the viral particle is intact, and stays in a latent state.

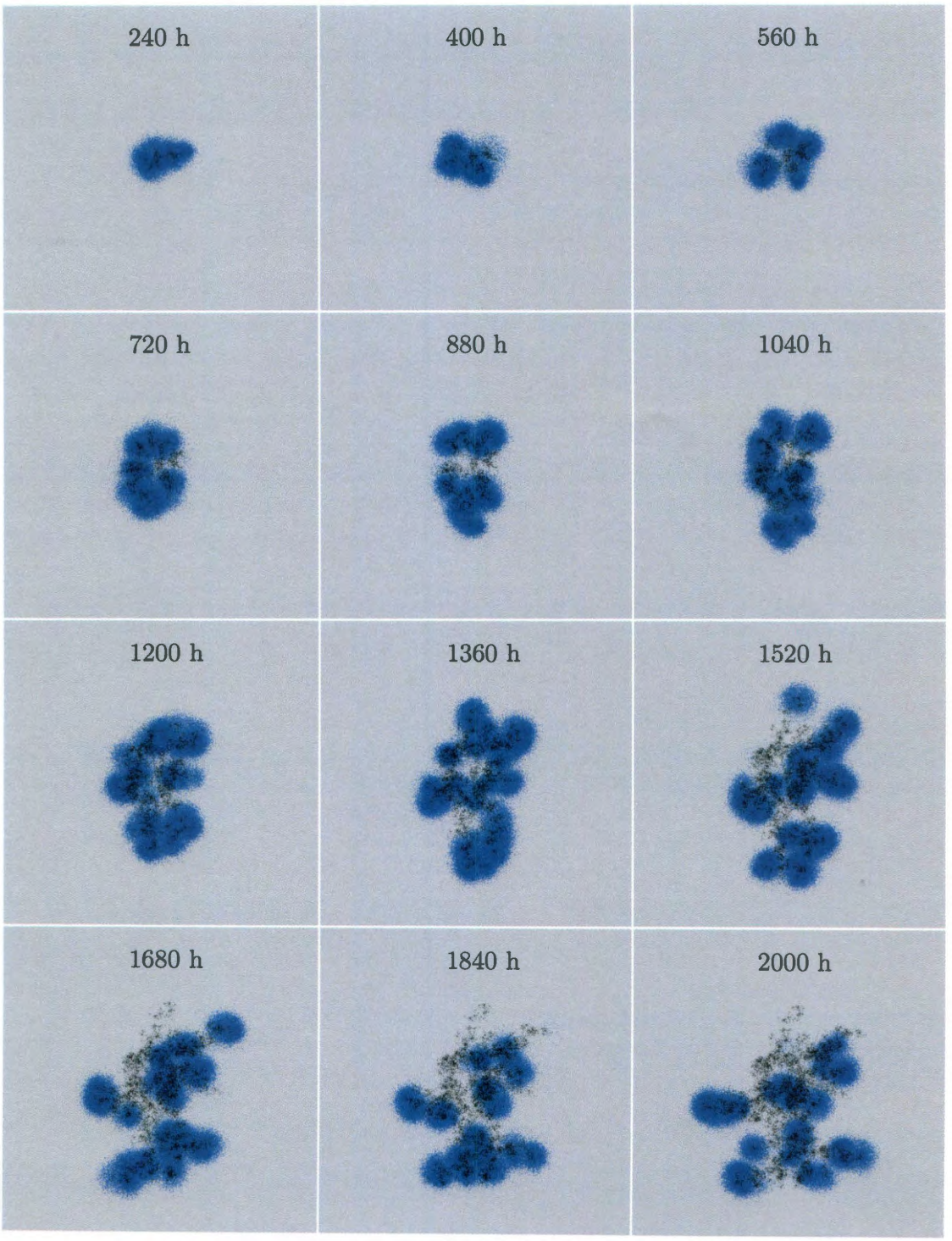


Figure 9.17 : Stochastic evolution of full model with infection rate = 1/(250 mol h)

9.9.0.2 Second stage of infection: viral genome release

Viral particle capsid opens releasing the viral genome, which uses the cell's machinery to produce viral double stranded mRNA intermediates and proteins.

$$\mathrm{IC}_1 \xrightarrow{\text{IC\_deterioration\_r}} \mathrm{IC}_2 \tag{9.17}$$

The intermediates are sensed by toll-like receptors and RIG-I proteins, starting an immune response that in the next two stages will produce interferon (see Equation 9.19 and 9.23).

9.9.0.3 Third stage of infection: cell releases interferon

The interferon gene is activated:

$$\mathrm{IC}_2 \xrightarrow{\text{IC\_deterioration\_r}} \mathrm{IC}_3 \tag{9.18}$$

and the cell starts to produce interferon molecules, releasing them into the medium:

$$\mathrm{IC}_3 \xrightarrow{\text{IC\_ifn\_production\_r}} \mathrm{IC}_3 + \mathrm{Ifn} \tag{9.19}$$

There is a chance that the cell will undergo apoptosis:

$$\mathrm{IC}_3 \xrightarrow{\text{IC\_death\_r / 4}} \mathrm{DC} \tag{9.20}$$

The cell continues to produce viral proteins and genomes.

Parameter	Value
UC_infection_r	1/(2.5 mol h)
UC_resistance_r	1/(50 mol h)
RC_recover_r	1/(24 h)
IC_deterioration_r	1/(3 h)
IC_ifn_production_r	1/(10 s)
IC_vir_production_r	1/(30 s)
IC_death_r	1/(1 h)
ifn_degr_r	1/(1.5 h)
vir_degr_r	1/(1 h)
D_ifn_lit	5.75×10^{-5} mm^2/s
D_vir_lit	2.326×10^{-6} mm^2/s

Table 9.1 : Parameters used in reference system. The values are suitable to be directly used in a grid of 100×100 compartments. For other grid densities, the top two constant rates need to be rescaled accordingly.

9.9.0.4 Fourth stage of infection: cell releases viral particles and interferon

Viral proteins and genomes have undergone the processes needed to assemble:

$$\mathrm{IC}_3 \xrightarrow{\text{IC\_deterioration\_r}} \mathrm{IC}_4 \tag{9.21}$$

and the cell starts releasing viral particles into the medium at half rate:

$$\mathrm{IC}_4 \xrightarrow{\text{IC\_vir\_production\_r / 2}} \mathrm{IC}_4 + \mathrm{Vir} \tag{9.22}$$

Viral proteins interfere with interferon production, that progresses at half rate:

$$\mathrm{IC}_4 \xrightarrow{\text{IC\_ifn\_production\_r / 2}} \mathrm{IC}_4 + \mathrm{Ifn} \tag{9.23}$$

There is a higher chance that the cell will undergo apoptosis:

$$\mathrm{IC}_4 \xrightarrow{\text{IC\_death\_r / 2}} \mathrm{DC} \tag{9.24}$$

The cell continues to produce viral proteins and genome.

9.9.0.5 Fifth stage of infection: cell releases viral particles and finally dies

Viral proteins shut down interferon production:

$$\mathrm{IC_4} \xrightarrow{\text{IC\_deterioration\_r}} \mathrm{IC_5} \tag{9.25}$$

Cell releases viral particles into the medium at full rate:

$$\mathrm{IC_5} \xrightarrow{\text{IC\_vir\_production\_r}} \mathrm{IC_5 + Vir} \tag{9.26}$$

Infected cell finally dies:

$$\mathrm{IC_5} \xrightarrow{\text{IC\_death\_r}} \mathrm{DC} \tag{9.27}$$

9.9.0.6 Cell resistance

Interferon molecules interact with uninfected cells in a competitive fashion with viral particles (see Equation 9.16).

$$\mathrm{Ifn + UC} \xrightarrow{\text{UC\_resistance\_r / ifn\_avg}} \mathrm{RC} \tag{9.28}$$

The interferon molecule is consumed in the process. We do not consider ulterior interactions between interferon molecules and cells at other states other than uninfected. Interferon molecules are eliminated from the system according to Equation 9.30.

Cell resistance is a temporary property that the cell looses at certain point:

$$\mathrm{RC} \xrightarrow{\text{RC\_recover\_r}} \mathrm{UC} \tag{9.29}$$

9.9.0.7 Interferon molecule degradation

Interferon molecules degrade:

$$\mathrm{Ifn} \xrightarrow{\text{ifn\_degr\_r}} \otimes \tag{9.30}$$

9.9.0.8 Viral particle degradation

Viral particles degrade:

$$\mathrm{Vir} \xrightarrow{\text{vir\_degr\_r}} \otimes \tag{9.31}$$

Chapter 10

Discussion

We are interested in the study of molecular cell biology systems, both in the cases where spatial effects can be neglected, and in the cases where it cannot.

The first part of this work concerns biological systems in which it is normal to assume that reactions happen in a container with constant volume where all reactants are perfectly mixed and in thermal equilibrium. In this formulation, each kind of reaction has a propensity function, or hazard h, associated. Propensity has a specific form that depends on the type of reaction considered, and is usually a function of the number of reacting species present at a given time.

Under such assumptions, the probability that a particular reaction will take place in a time interval $[t, t + \Delta t]$, is equal to $h\Delta t + o(\Delta t)$, and the time to such a reaction, in the absence of other reactions taking place, is exponentially distributed with parameter h. This stochastic process is a time-continuous Markov chain.

It is customary, under the additional assumption that the number of molecules of each reacting species is sufficiently high, to model such a system using a set of ordinary differential equations (ODEs). Given initial conditions, this system can be solved, in most cases only numerically, and its solutions are interpreted as the time evolutions of the expected values of the species of the associated stochastic process. However, eukaryotic cells have at most two gene copies, with corresponding mRNA levels in the order of $10^2 - 10^3$ molecules. In prokaryotic cells, mRNA molecules are present in low numbers and most of them are unstable [67]. As a consequence, the assumption of high number of reacting molecules of each type seldom applies in the biological systems we study.

In the introductory Chapter 2 we discussed conditions under which the solution

of a system of ordinary differential equations coincides with the expected value of the original stochastic kinetic model. The conclusion is that both approaches coincide if *all* reactions have either zero or first order mass action kinetics, regardless of the number of molecules involved.

However, in most but the simplest biological networks, it is common to find reactions that are of second order mass action kinetics. Examples include: binding of transcription factors and repressors to genes, dimerization of proteins, formation of complexes, catalytic action, phosphorylation, signal molecules binding to receptors, or viral particles infecting a cell. There also are cases in which nonlinear functions, such as Michaelis-Maenten function, are used. Reactions of order higher than two are seldom used and can be, perhaps more correctly, modeled as a succession of second order reactions.

In Chapter 4 we presented an original software tool, named bioPN (biological Petri Nets), that allows, given a set of coupled chemical reactions represented as Petri nets, and initial conditions, to numerically solve the resulting system of ODEs, and to simulate, using a range of methods, stochastic trajectories.

After validating the algorithms against theoretical results, we used the tool to compare deterministic and stochastic results in simple systems. We started from an unregulated network composed of a gene, which transcribes mRNA, which translates a protein. As all the reactions involved were first order mass action kinetics, we verified the agreement of the results of both methods.

However, when inhibition of the gene by its own protein was considered, the agreement between both methods ended. Varying the mRNA translation and degradation rates but keeping their ratio constant, we observed discrepancy in values by a factor of four in the time evolution of the protein. The deterministic solution under-estimates the expected value of the stochastic process. After further modification of the model by introducing dimerization of the protein and replacing the repressing protein by its

dimer, we observed, after tuning mRNA rates as described above, three different behaviors in the time solution of the dimer: 1) the deterministic solution over-estimating the stochastic mean, 2) agreement between both, and 3) the deterministic solution under-estimating the stochastic mean.

We conclude Chapter 4 by showing results, already published [58], of the effect of bi-stability that the deterministic solution failed to capture.

In Chapter 5 we presented current work on a model of cellular sensing of double stranded RNA, a joint effort with researchers at the University of Texas Medical Branch and Institute of Fundamental Technological Research, Polish Academy of Sciences, Warsaw, Poland. As this work involves a deterministic model, we compared the results with its stochastic counterpart. As expected, in the cases of regulated genes, discrepancies between the two approaches are observed, both in levels and in dynamics of different constituents. However, during the first hours after dsRNA electroporation, both systems react similarly. This effect may be related to the instantaneous introduction of 10^5 dsRNA molecules, leading to a strong response that overcomes any other effects.

The second part of the thesis is devoted to systems in which spatial effects cannot be neglected. We review the relevant mathematical background for deterministic and stochastic reaction-diffusion systems in Chapter 6. In Chapter 7 we presented sbioPN, an extension of bioPN to spatial systems. In compartment-based approach adopted in sbioPN, diffusion is modeled as a series of first order reactions. Therefore, in a purely diffusive model, or in a reaction-diffusion model where all reactions are zero or first order mass kinetics, the deterministic evolution should agree with the averages of stochastic runs. This has been verified.

In Chapter 8 we presented the stochastic extension of a one-dimensional reaction-diffusion model of early carcinogenesis that exhibits Turing instabilities in its deterministic formulation. We compared two hundred and fifty cases of each system by

modifying three parameters: 1) the number of peaks of the cosine perturbation of the spatially homogeneous steady state, which constituted the initial conditions, 2) the diffusion coefficient, and 3) the grid density.

These parameters only affect the spatial system. The model without diffusion exhibits a four fold discrepancy in the time evolution of pre-cancerous cells and bound growth factor. The reaction-diffusion system, as a consequence, shows a similar effect: no stochastic system could keep, when averaged over a large number of realizations, the initial levels of the spikes generated by the corresponding deterministic system; all the stochastic cases exhibited a sudden drop in spike levels.

Diffusion and grid density play a role in the ability of the stochastic system to maintain, when averaged over a large number of realizations, the initial structure of spikes after the drop. Higher diffusion coefficients favor the preservation of the initial spikes. Moreover, for $\gamma = 1$ and $\gamma = 10$, the stochastic system maintains spike levels indefinitely. Regarding grid density, when using a diffusion coefficient too low to maintain spike levels ($\gamma = 100$), coarser grids make the decay more gradual whereas denser grids make it more steep.

Finally, in Chapter 9 we studied a two-dimensional reaction-diffusion model of viral infection and interferon production, and compared the deterministic runs to single realizations of the stochastic version. The results are interesting. On one hand, the deterministic model frequently evolves faster than the stochastic model. On the other hand, rich dynamics of the stochastic system are not always captured by the deterministic evolution. Systems without diffusion show analogous discrepancies.

Based on these examples, we conclude that deterministic and stochastic approaches seldom agree, at least in cases that explore dynamics in regulated biological systems, unless unrealistically high numbers of molecules are used. The disagreement between both approaches is manifested in both reaction-only and reaction-diffusion systems.

Appendix A

Appendix for Part 1

A.1 Chapter 2

Petri net graphical representations for eactions 2 to 8 of the model defined in Chapter 2 are presented in Figures A.1 to A.7.

$$\mathrm{Ga} + \mathrm{Dimer} \xrightarrow{h_2} \mathrm{Gi}$$

$$\mathrm{Pre}[2,] = (0 \quad 1 \quad 0 \quad 0 \quad 1)$$
$$\mathrm{Post}[2,] = (1 \quad 0 \quad 0 \quad 0 \quad 0)$$

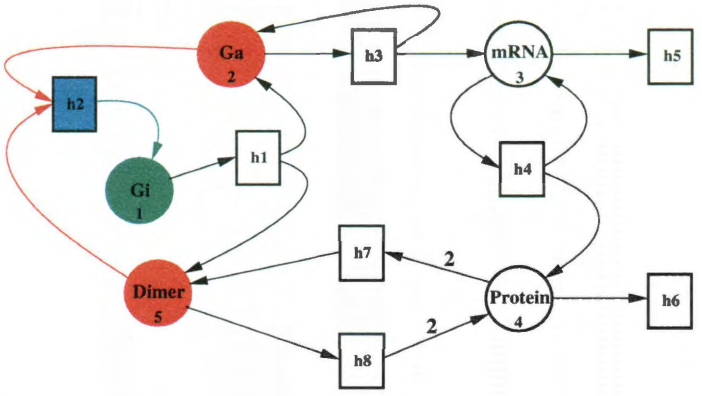


Figure A.1 : Reaction 2: Gene inactivation.

$$Ga \xrightarrow{h_3} Ga + mRNA$$

$$Pre[3,] = (0 \quad 1 \quad 0 \quad 0 \quad 0)$$
$$Post[3,] = (0 \quad 1 \quad 1 \quad 0 \quad 0)$$

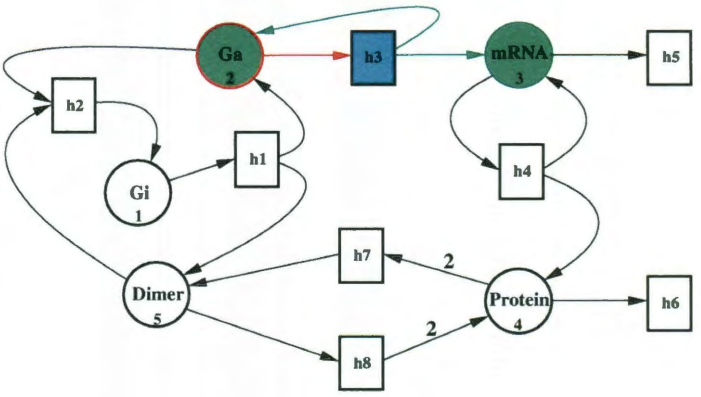


Figure A.2 : Reaction 3: Transcription.

$$mRNA \xrightarrow{h_4} mRNA + Protein$$

$$Pre[4,] = (0 \quad 0 \quad 1 \quad 0 \quad 0)$$
$$Post[4,] = (0 \quad 0 \quad 1 \quad 1 \quad 0)$$

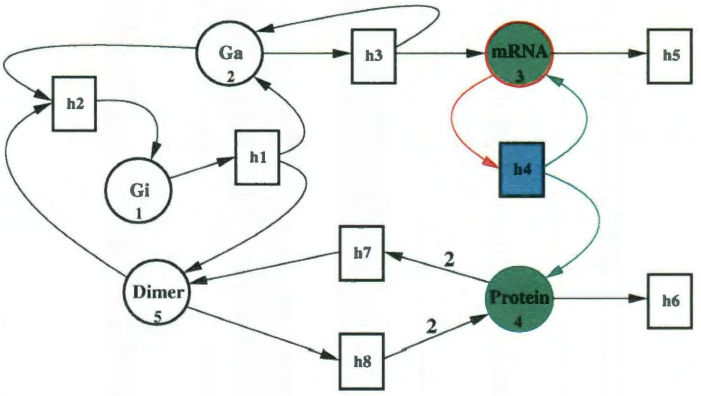


Figure A.3 : Reaction 4: Translation.

$$mRNA \xrightarrow{h_5} \otimes$$

$$Pre[5,] = (0 \quad 0 \quad 1 \quad 0 \quad 0)$$
$$Post[5,] = (0 \quad 0 \quad 0 \quad 0 \quad 0)$$

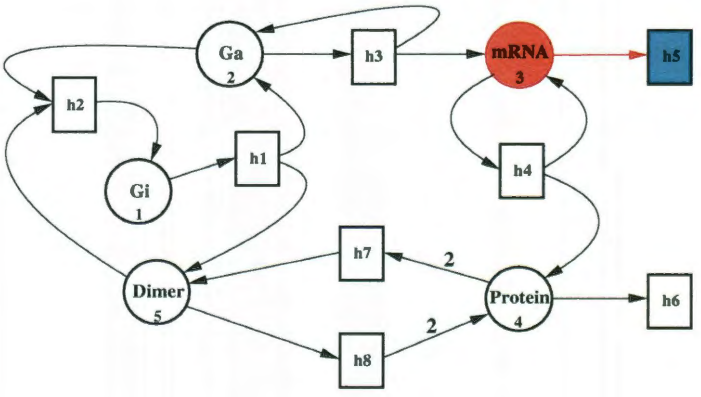


Figure A.4 : Reaction 5: mRNA Degradation.

$$\text{Protein} \xrightarrow{h_6} \otimes$$

$$\text{Pre}[6,] = (0 \quad 0 \quad 0 \quad 1 \quad 0)$$
$$\text{Post}[6,] = (0 \quad 0 \quad 0 \quad 0 \quad 0)$$

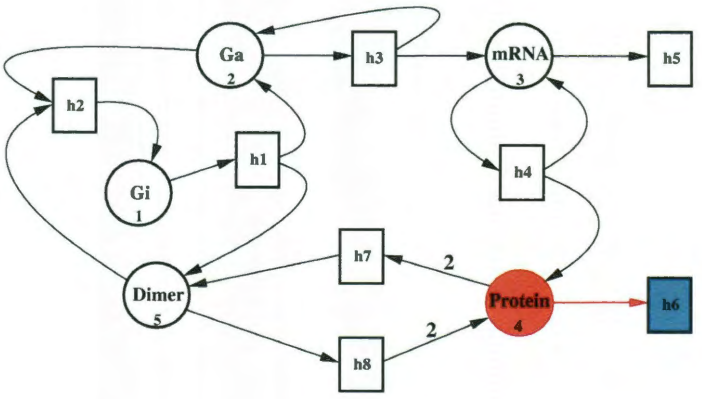


Figure A.5 : Reaction 6: Protein Degradation.

$$2\ \text{Protein} \xrightarrow{h_7} \text{Dimer}$$

$$\text{Pre}[7,] = (0 \quad 0 \quad 0 \quad 2 \quad 0)$$
$$\text{Post}[7,] = (0 \quad 0 \quad 0 \quad 0 \quad 1)$$

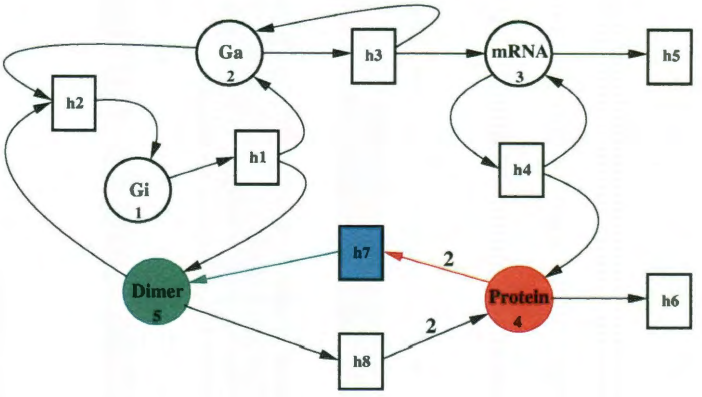


Figure A.6 : Reaction 7: Protein Dimerization.

$$\text{Dimer} \xrightarrow{h_8} 2\ \text{Protein}$$

$$\text{Pre}[8,] = (0 \quad 0 \quad 0 \quad 0 \quad 1)$$
$$\text{Post}[8,] = (0 \quad 0 \quad 0 \quad 2 \quad 0)$$

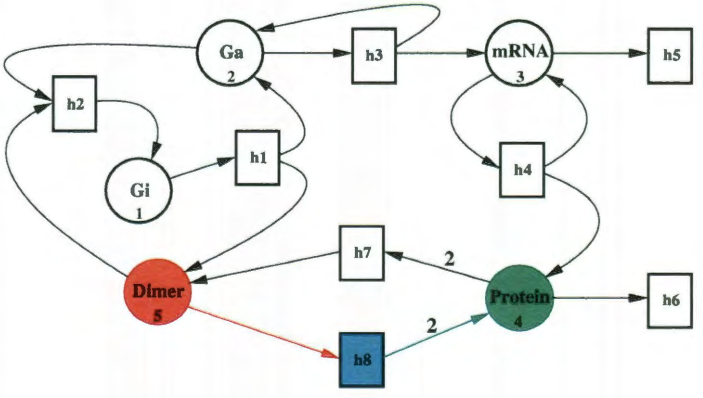


Figure A.7 : Reaction 8: Dimer Dissociation.

A.1.1 Derivation of Equation 2.31

This derivation can be found on Wilkinson [69] (page 159).

$$
\begin{aligned}
\frac{d}{dt}\mathrm{E}\left[X_t\right] &= \frac{d}{dt}\sum_{x\in\mathcal{M}} x\; p(x,t) = \sum_{x\in\mathcal{M}} x\; \frac{d}{dt}p(x,t) \\
&= \sum_{x\in\mathcal{M}}\left\{ x \sum_{i=1}^{v}\left[h_i(x-S^{(i)},c_i)\; p(x-S^{(i)},t) - h_i(x,c_i)\; p(x,t)\right]\right\} \\
&= \sum_{i=1}^{v}\left\{ \sum_{x\in\mathcal{M}}\left[x\; h_i(x-S^{(i)},c_i)\; p(x-S^{(i)},t)\right] - \sum_{x\in\mathcal{M}}\left[x\; h_i(x,c_i)\; p(x,t)\right]\right\} \\
&= \sum_{i=1}^{v}\left\{ \sum_{x\in\mathcal{M}}\left[(x+S^{(i)})\; h_i(x,c_i)\; p(x,t)\right] - \sum_{x\in\mathcal{M}}\left[x\; h_i(x,c_i)\; p(x,t)\right]\right\} \\
&= \sum_{i=1}^{v}\left\{ \mathrm{E}\left[(X_t+S^{(i)})\; h_i(X_t,c_i)\right] - \mathrm{E}\left[X_t\; h_i(X_t,c_i)\right]\right\} \\
&= \sum_{i=1}^{v}\mathrm{E}\left[S^{(i)}\; h_i(X_t,c_i)\right] \\
&= \sum_{i=1}^{v} S^{(i)}\;\mathrm{E}\left[h_i(X_t,c_i)\right]
\end{aligned}
$$

A.2 Chapter 4

A.2.1 Hybrid simulations. Stochastic: reactions that modify gene and mRNA. Deterministic: reactions that modify protein

The variance of mRNA is the same as the one on the previous subsection. The variance of Protein is given by [53]:

$$
\begin{aligned}
\mathrm{Var}\left[\mathrm{Protein}\right] =& rb\frac{1+c+b+r}{c(1+r)(1+c+b)(r+c+b)}\mathrm{E}^2\left[\mathrm{Protein}\right] + \\
& \frac{r}{1+r}\frac{\mathrm{E}^2\left[\mathrm{Protein}\right]}{\mathrm{E}\left[\mathrm{mRNA}\right]}
\end{aligned}
\tag{A.1}
$$

This case was used to validate the function HaseltineRawlings. Results are shown in Figure A.8.

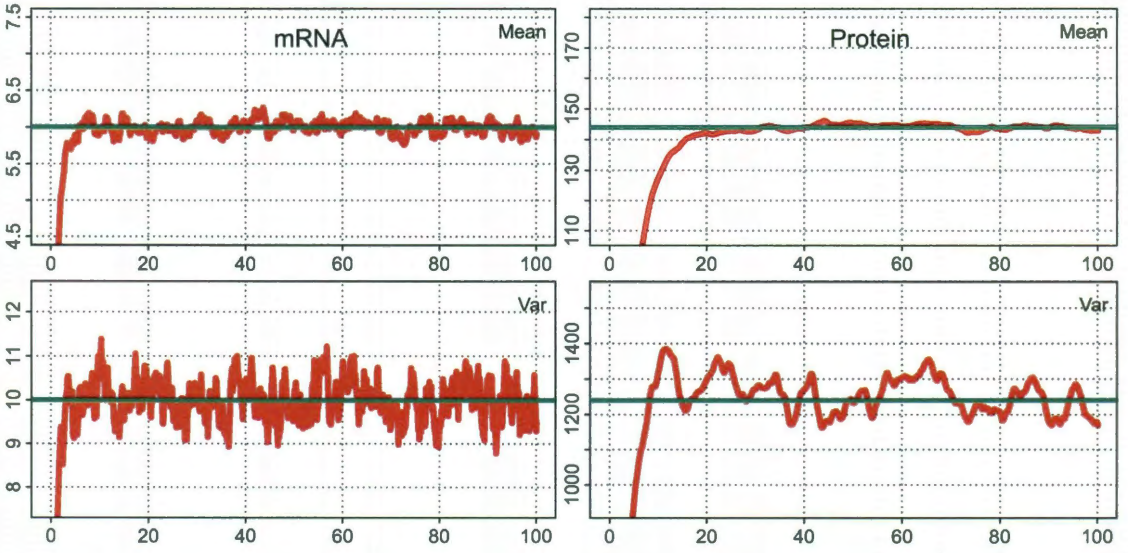


Figure A.8 : Hybrid: Gene activation/inactivation and mRNA reactions stochastic, protein reactions deterministic. Horizontal axis: time units. Vertical axis: number of units. Green lines: Theoretical results. Red lines: Simulations using bioPN (1000 runs).

A.2.2 Hybrid simulations. Stochastic: reactions that modify gene. Deterministic: reactions that modify protein and mRNA

The variance of mRNA and Protein is given by [53]:

$$\text{Var}\left[\text{mRNA}\right] = \frac{b}{c(1+c+b)}\text{E}^2\left[\text{mRNA}\right] \tag{A.2}$$

$$\text{Var}\left[\text{Protein}\right] = rb\frac{1+c+b+r}{c(1+r)(1+c+b)(r+c+b)}\text{E}^2\left[\text{Protein}\right] \tag{A.3}$$

This case was also used to validate the function HaseltineRawlings. Results are shown in Figure A.9.

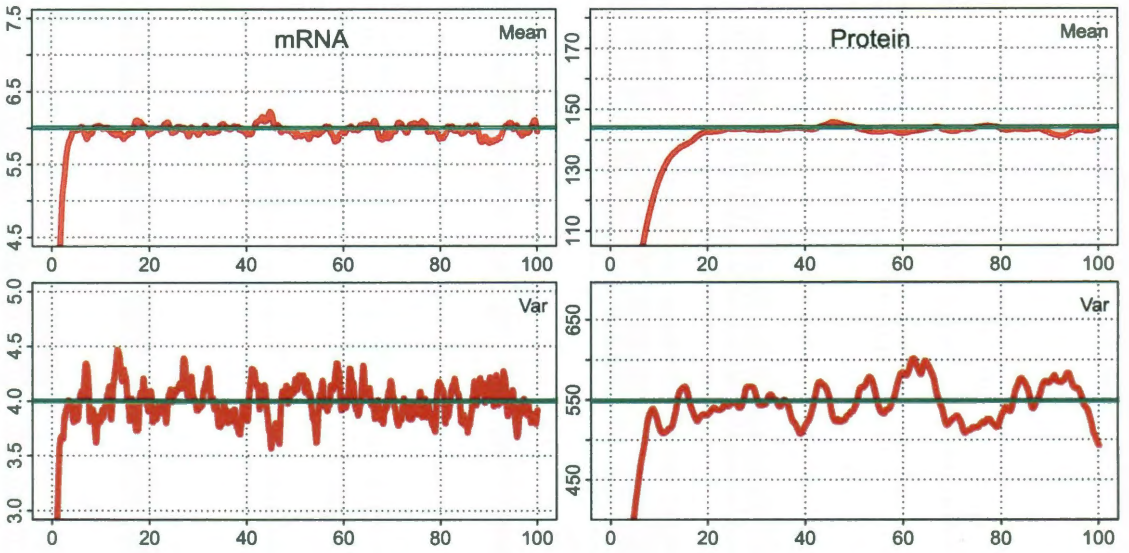


Figure A.9 : Hybrid: Gene activation/inactivation stochastic, mRNA and protein reactions deterministic. Horizontal axis: time units. Vertical axis: number of units. Green lines: Theoretical results. Red lines: Simulations using bioPN (1000 runs).

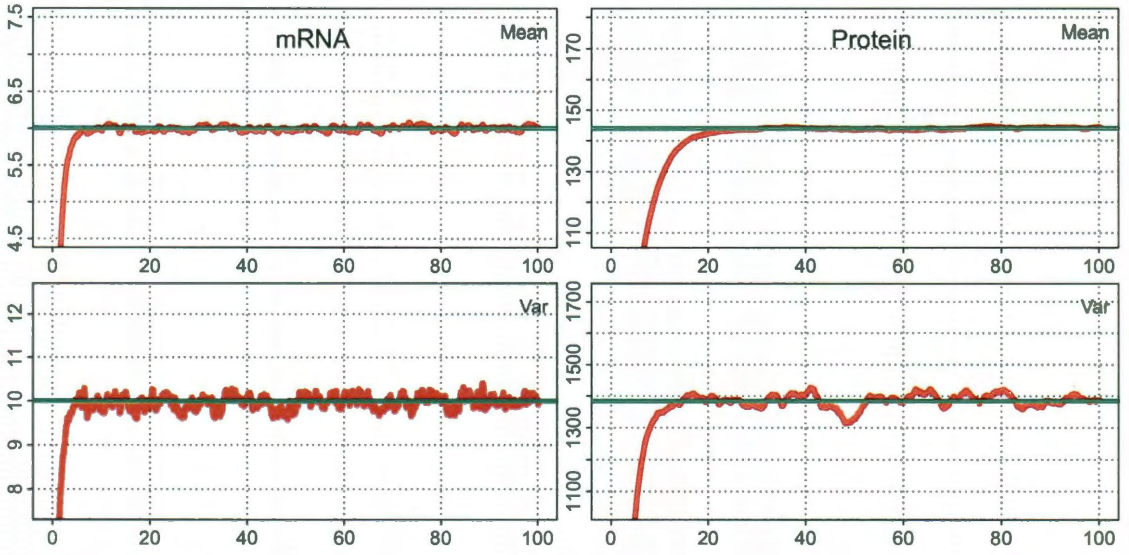


Figure A.10 : All reactions stochastic. Horizontal axis: time units. Vertical axis: number of units. Green lines: Theoretical results. Red lines: Simulations using bioPN (10,000 runs).

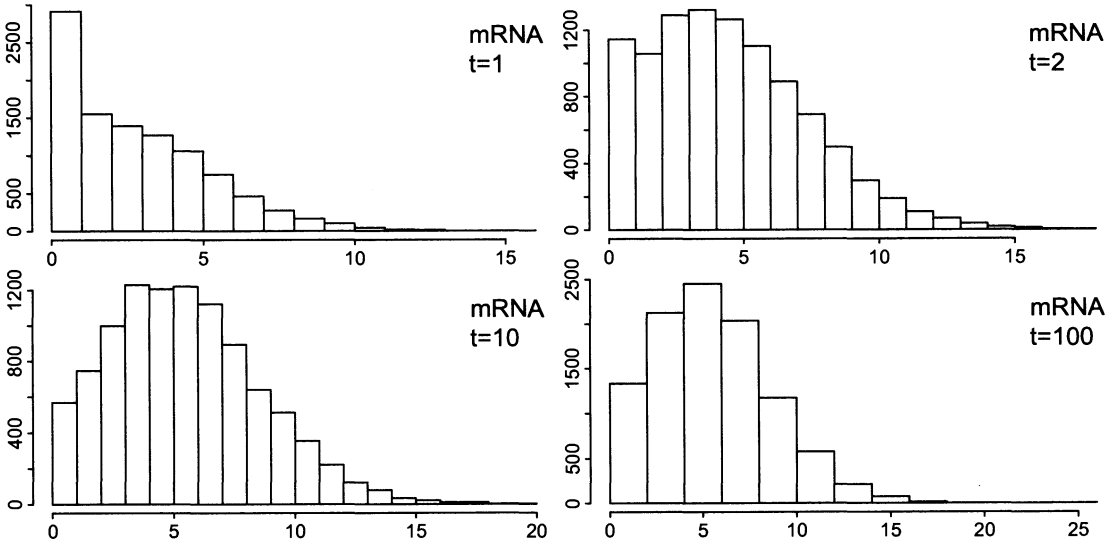


Figure A.11 : All reactions stochastic. Frequency histogram: mRNA

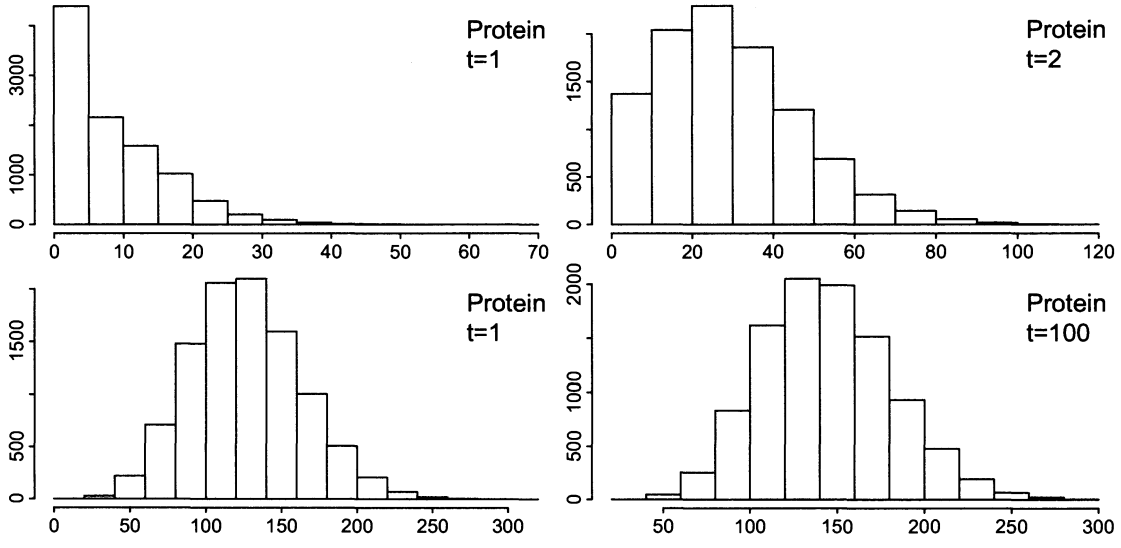


Figure A.12 : All reactions stochastic. Frequency histogram: Protein

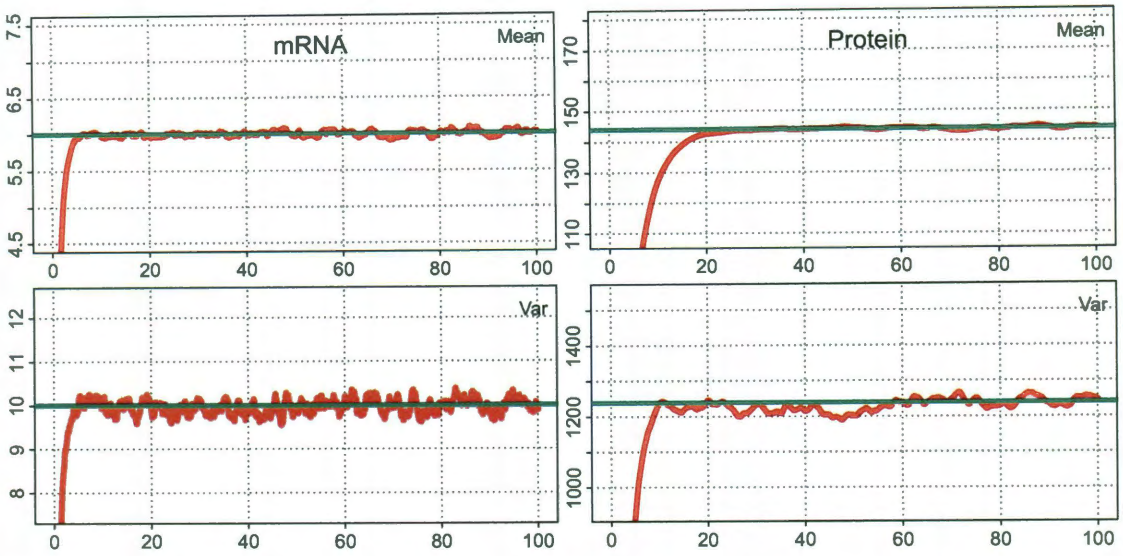


Figure A.13 : Hybrid: Gene activation/inactivation and mRNA reactions stochastic, protein reactions deterministic. Horizontal axis: time units. Vertical axis: number of units. Green lines: Theoretical results. Red lines: Simulations using bioPN (10,000 runs).

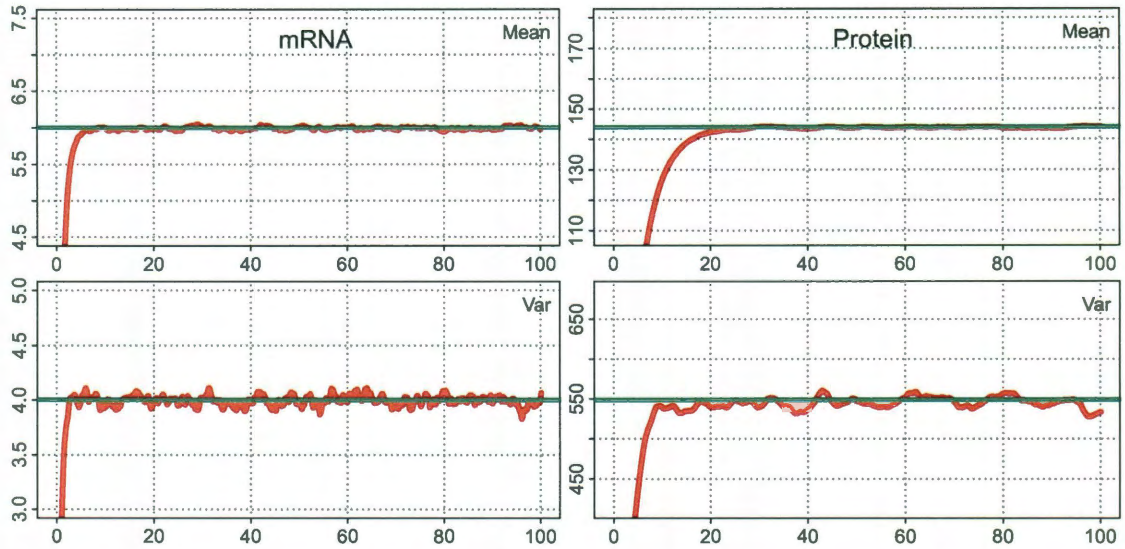


Figure A.14 : Hybrid: Gene activation/inactivation stochastic, mRNA and protein reactions deterministic. Horizontal axis: time units. Vertical axis: number of units. Green lines: Theoretical results. Red lines: Simulations using bioPN (10,000 runs).

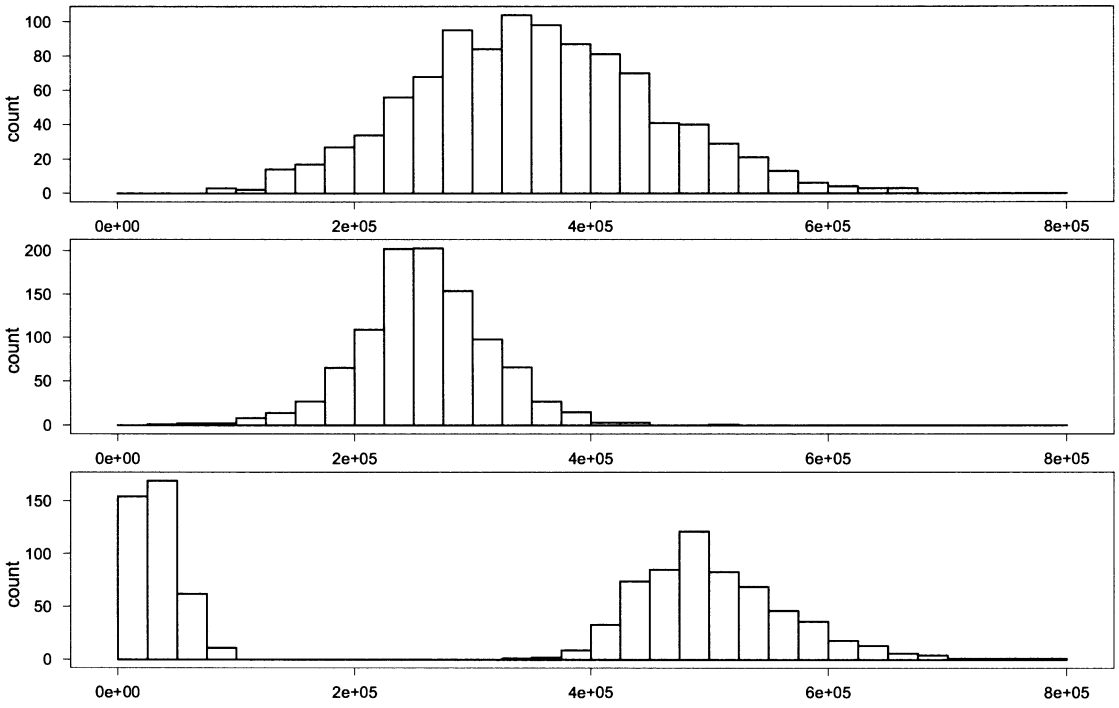


Figure A.15 : Histograms of phosphorylated nuclear P53 at 35, 50, and 80 hours, respectivelly.

Appendix B

bioPN manual

B.1 Availability

Updated versions of the manual, as well as the code needed to run the examples, and the library (currently version 1.0.0), can be found on http://www.stat.rice.edu/~mathbio/bioPN.

B.2 Functions in the Library

The library has 4 kind of functions:

- Exact stochastic simulation (4 functions)
  - GillespieOptimDirect: "in house" optimized version of the Gillespie SSA algorithm.
  - GillespieDirectGB: optimized version of the Gillespie SSA algorithm according to Gibson and Bruck [18]
  - `GibsonBruck`: Next reaction method by Gibson and Bruck [18].
  - `GillespieDirectCR`: constant-time composition/rejection algorithm [64].
- Pure deterministic integration (1 function)
  - `RungeKuttaDormandPrince45`: Runge-Kutta Dormand and Prince algorithm
- A hybrid of the above (1 function)
  - `HaseltineRawlings`: Haseltine and Rawlings method with deterministic integration of fast transitions [27]
- Dynamic re-partitioning algorithm (1 function)
  - `PartitionedLeaping`: Partitioned leaping algorithm (PLA) [24]. Transitions are dynamically re-partitioned between 4 categories (very slow, slow, medium, and fast) and simulated accordingly (SSA, τ-leaping, CLE, deterministic).

B.2.1 Usage

```
GillespieOptimDirect(model, timep, delta = 1, runs = 1)

GillespieDirectGB(model, timep, delta = 1, runs = 1)

GibsonBruck(model, timep, delta = 1, runs = 1)

GillespieDirectCR(model, timep, delta = 1, runs = 1)


RungeKuttaDormandPrince45(model, timep, delta = 1, ect = 1e-09)


HaseltineRawlings(model, timep, delta = 1, runs = 1, ect = 1e-09)


PartitionedLeaping(model, timep, delta = 1, runs = 1)
```

B.2.2 Arguments

- `model`: list containing the following named elements:
  - `pre`: pre matrix, with as many rows as transitions (reactions), and columns as places (reactants). It has the stoichiometrics of the left sides of the reactions.
  - `post`: post matrix, with as many rows as transitions, and columns as places (products). It has the stoichiometrics of the right sides of the reactions.
  - `h`: list of propensity constants or functions returning the propensity (with as many elements as transitions).
  - `slow`: vector of zeros for slow transitions and ones for fast transitions. Only needed for `HaseltineRawlings`. Ignored otherwise.
  - `M`: Initial marking (state) of the system.
  - `place`: vector with names of the places.
  - `transition`: vector with names of the transitions.
- `timep`: It can be either a numeric, indicating for how long (in the same time units as the propensity constants) the process will run, or a functions (R or C),

in which case can be used to change the protocol at time intervals. See below.

- `delta`: Interval time at which the state will be saved.
- `runs`: How many runs will be performed.
- `ect`: Precision for the fast reactions.

B.2.3 Returned Value

The functions return a list with the following elements:

- `place`: vector with the names of the places if supplied. If not, the function creates names as follows: P1, P2, ...
- `transition`: vector with the names of the transitions if supplied. If not, the function creates names as follows: T1, T2, ...
- `dt`: vector containing the discretized times at which the state is saved (according to delta)
- `run`: list with as many elements as runs. We will describe the first element, run[[1]], as the rest have exactly the same structure. It is also a list, with the following elements:
  - `run[[1]]$M`: list with as many elements as places, each of them containing the state of the system sampled according to `delta`.
  - `run[[1]]$transitions`: vector with as many elements as transitions, with the total of time each slow reaction fired.
  - `run[[1]]$tot.transitions`: numeric with the summ of `run[[1]]$transitions`.

B.3 Example: Stochasticity in Gene Regulation

The first example is a simple model that has steady state theoretical solutions [53].

B.3.1 Reactions

$$I \xrightarrow{c} A \qquad mRNA \xrightarrow{K} mRNA + protein$$

$$A \xrightarrow{b} I \qquad mRNA \xrightarrow{1} \otimes$$

$$A \xrightarrow{H} A + mRNA \qquad protein \xrightarrow{r} \otimes$$

This model consists in 4 places (reactants, products) and 6 transitions (reactions). They are illustrated in the following table

Places	Transitions
1. inactive gene I	1. gene activation
2. active gene A	2. gene inactivation
3. $mRNA$	3. transcription
4. *protein*	4. mRNA degradation
	5. translation
	6. protein degradation

B.3.2 Model Definition: First Approach

B.3.2.1 Pre Matrix

The first needed structure is the 6×4 matrix *pre*, which accounts for the stoichiometries of the left sides of the reactions.

```
pre <- matrix(c(1,0,0,0,
                0,1,0,0,
                0,1,0,0,
                0,0,1,0,
                0,0,1,0,
                0,0,0,1),
              ncol=4,byrow=T)
```

Transitions (reactions) are associated with rows, while places (reactants, products) are associated with columns.

B.3.2.2 Post Matrix

Define *post*, to account for the stoichiometries of the right side of the reactions.

```
post <- matrix(c(0,1,0,0,
                 1,0,0,0,
                 0,1,1,0,
                 0,0,0,0,
                 0,0,1,1,
                 0,0,0,0),
               ncol=4,byrow=T)
```

B.3.2.3 Propensities

Define the vector of propensities or hazards

```
h <- list(c=3, b=2, H=10, 1, K=6, r=0.25)
```

B.3.2.4 Initial Conditions

```
M <- c(1,0,0,0)
```

B.3.2.5 Fast and Slow Transitions

Only necessary for `HaseltineRawlings`. Ignored by the other functions.

```
slow <- c(1,1,0,0,0,0)
```

This case will treat gene inactivation/activation as slow, while the rest as fast.

B.3.2.6 Model Definition

```
model <- list(pre=pre, post=post, h=h, M=M, slow=slow)
```

B.3.2.7 Load the Library

Load the library (can be done at any point before the call to the functions)

```
library(bioPN)
```

B.3.2.8 Calling the Functions

```
timep <- 200
delta <- 1

## Completely deterministic run
Sim <- RungeKuttaDormandPrince45(model, timep, delta)

runs <- 10000
## Completely stochastic run
set.seed(19761111)  ## Set a seed
Sim <- GillespieOptimDirect(model, timep, delta, runs)
#Sim <- GibsonBruck(model, timep, delta, runs)
#Sim <- GillespieDirectGB(model, timep, delta, runs)
#Sim <- GillespieDirectCR(model, timep, delta, runs)

## Hybrid run
set.seed(19761111)  ## Set a seed
Sim <- HaseltineRawlings(model, timep, delta, runs)
```

B.3.3 Model Definition: Second Approach

This approach is closer to the way reactions are defined. It uses the functions in the file "helper.R" provided in the section devoted to R code. Other approaches may be implemented in the future. The model has to be defined in the main body (not inside a function), and its use has the side effect of generating three name variables: model, L (for left side of the reaction), and R (for right side of the reaction), which are used as part of the definition. It will also create variable names for the defined places.

B.3.3.1 Names of Places and engine start

```
source("helper.R")

####### Constants definition (convenient but not required)
H <- 10
K <- 6
r <- 0.25
c <- 3
b <- 2
#######

place <- c( "Gi", "Ga", "mRNA", "Protein")

init(place)  ## This will create the structures model, L, and R
             ## and variables for each place.
```

B.3.3.2 Reaction (transition) definition

```
## Gi -> Ga
h <- c
L[Gi] <- 1
R[Ga] <- 1
atr() ## Add this reaction

## Ga -> Gi
h <- b
L[Ga] <- 1
R[Gi] <- 1
atr()

## Ga -> Ga + mRNA
h <- H
L[Ga] <- 1
R[Ga] <- 1; R[mRNA] <- 1
atr()

## mRNA -> mRNA + Protein
h <- K
L[mRNA] <- 1
R[mRNA] <- 1; R[Protein] <- 1
atr()

## mRNA -> 0
h <- 1
L[mRNA] <- 1
atr()

## Protein -> 0
h <- r
L[Protein] <- 1
atr()
```

B.3.3.3 Initial Conditions

```
model$M <- rep(0, model$places)
model$M[Gi] <- 1
```

B.3.3.4 Calling the Functions

```
##############################
## Completely Deterministic ##
##############################
Sim <- RungeKuttaDormandPrince45(model, timep, delta, ect)

############################
## Completely Stochastic ##
############################
set.seed(19761111)
Sim <- GillespieOptimDirect(model, timep, delta, runs)
```

```
Sim <- GibsonBruck(model, timep, delta, runs)
Sim <- GillespieDirectGB(model, timep, delta, runs)
Sim <- GillespieDirectCR(model, timep, delta, runs)

#####################################################################
## Hybrid: mRNA and protein fast, gene activation/inactivation slow ##
#####################################################################
slow <- rep(0, transitions)
slow[gene_activation] <- 1
slow[gene_inactivation] <- 1
model$slow <- slow

set.seed(19761111)
Sim <- HaseltineRawlings(model, timep, delta, runs, ect)
```

B.3.3.5 Example of Value Returned by the Functions

```
> str(Sim)
 $ place     : chr [1:4] "I" "A" "mRNA" "protein"
 $ transition: chr [1:6] "gene_activation" "gene_inactivation" "
    transcription" "mRNA_degradation" ...
 $ dt        : num [1:201] 0 1 2 3 4 5 6 7 8 9 ...
 $ run       :List of 10000
  ..$ :List of 3
  .. ..$ M             :List of 4
  .. .. ..$ : num [1:201] 1 0 1 0 0 0 0 0 0 0 ...
  .. .. ..$ : num [1:201] 0 1 0 1 1 1 1 1 1 1 ...
  .. .. ..$ : num [1:201] 0 2 2 2 7 6 10 10 6 9 ...
  .. .. ..$ : num [1:201] 0 2 18 16 48 82 105 142 147 168 ...
  .. ..$ transitions    : int [1:6] 245 245 1201 1192 6696 6540
  .. ..$ tot.transitions: int 16119
  ..$ :List of 3
  .. ..$ M             :List of 4
  .. .. ..$ : num [1:201] 1 0 0 0 0 1 1 0 1 0 ...
  .. .. ..$ : num [1:201] 0 1 1 1 1 0 0 1 0 1 ...
  .. .. ..$ : num [1:201] 0 1 2 5 6 5 4 8 10 10 ...
  .. .. ..$ : num [1:201] 0 7 18 39 47 81 88 92 103 128 ...
  .. ..$ transitions    : int [1:6] 243 242 1217 1205 7290 7156
  .. ..$ tot.transitions: int 17353
..... rest of the 10,000 runs follows...
```

The above result is from a completely stochastic run.

B.3.4 How to assign a function to a propensity

All the reactions in this example are mass-action ones, so the function calculates the corresponding propensity on the fly. However, there are cases where specific propensities need to be provided. Using as an example the first reaction (gene activation) the corresponding propensity is $c \times I$.

B.3.4.1 R function

```
h <- function() {c*y[I]}
```

NOTE: Declaring a function in R will result in a sensible performance degradation.

B.3.4.2 C function

```
  h <- load.cfn(place, paste("
#define c 3.
double cfn(double time, double *y) {
  return c*y[Gi];
}"))
```

B.3.5 How to manage time with a function

If the simulation needs to be separated into different sections (for example the first 50 seconds with one behaviour, and the second 150 with another), instead of assigning T a number, it can be assigned a function (R or C).

B.3.5.1 R function

```
timep <- function() {
    StartTime = round(StartTime,0)
    EndTime = 200
    if (StartTime == 0) {
        y[I] = 5
        EndTime = 50
    } else if (StartTime == 50) {
        y[I] = 10
    }
    list(EndTime,y)
}
```

B.3.5.2 C function

```
  h <- load.cfn(place, paste("
#include <Rmath.h>

double cfn(double dStartTime, double *y) {
  dStartTime = fround(dStartTime,0);
  if (dStartTime == 0) {
    y[I] = 5;
    return 50;
  } else if (dStartTime == 50) {
```

```
10      y[I] = 10;
11    }
12    return 200;
13
14  }"))
```

B.3.6 Removing temporary C files

After running the simulations, use the following to unload c objects created and delete the temporary files generated.

```
1   unload.cfns()
```

Appendix C

Additional material for Chapter 5

Figures C.1 to C.4 are interpreted as follows: each figure represents the knock-down of one of the target genes and the time series of its effect is shown. Blue lines are simulations of model without knock-down. Black dots are the observed experimental fold increase/decrease relative to blue lines. Dotted red lines are simulation of knock-down experiment. The closer red lines are to black dots, the better the fit.

C.1 Methodology used for mRNA experiments

On experiments 1 and 3, the harvested RNA was reversely transcribed into corresponding cDNA and run for quantitative real-time PCR (Q-RT-PCR) to examine the target gene expression level. Q-RT-PCR produces a curve showing the changes in the amount of product with each cycle. The cycle threshold (Ct) is used to quantify the amount of transcript, that is, where the amount of product is above background and in the linear range of PCR amplification. Relative changes in mRNA were calculated, normalizing to Glyceraldehyde 3-phosphate dehydrogenase (GAPDH) signal in each well using the delta-delta-Ct ($\Delta\Delta$Ct) algorithm [42]: first, the amount of target transcript is normalized to the amount of internal GAPDH control in each plate (ΔCt); second, the difference between the dsRNA "infection" (time1) and "uninfected" (time0), is calculated by:

$$\text{Fold change} = 2^{-(\Delta Ct_{\text{time1}} - \Delta Ct_{\text{time0}})} \tag{C.1}$$

For each target, a standard curve was generated using serial dilution of a plasmid containing the same PCR region as that measured by Q-RT-PCR.

For absolute quantification, the following protocol by Bustin [6] was used:

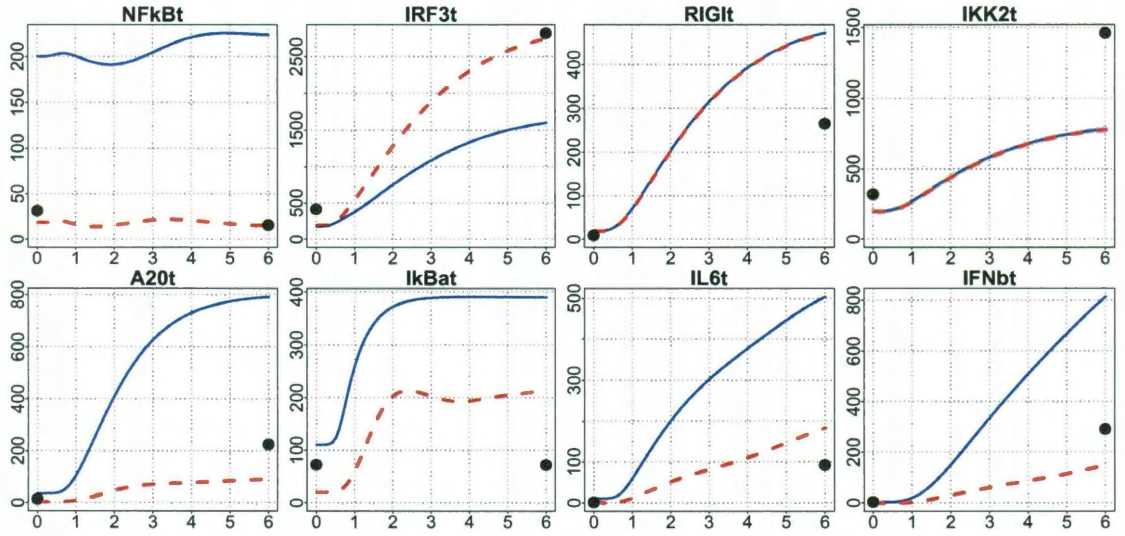


Figure C.1 : Simulation results: NF-κB Knock-down, compared to experimental results.

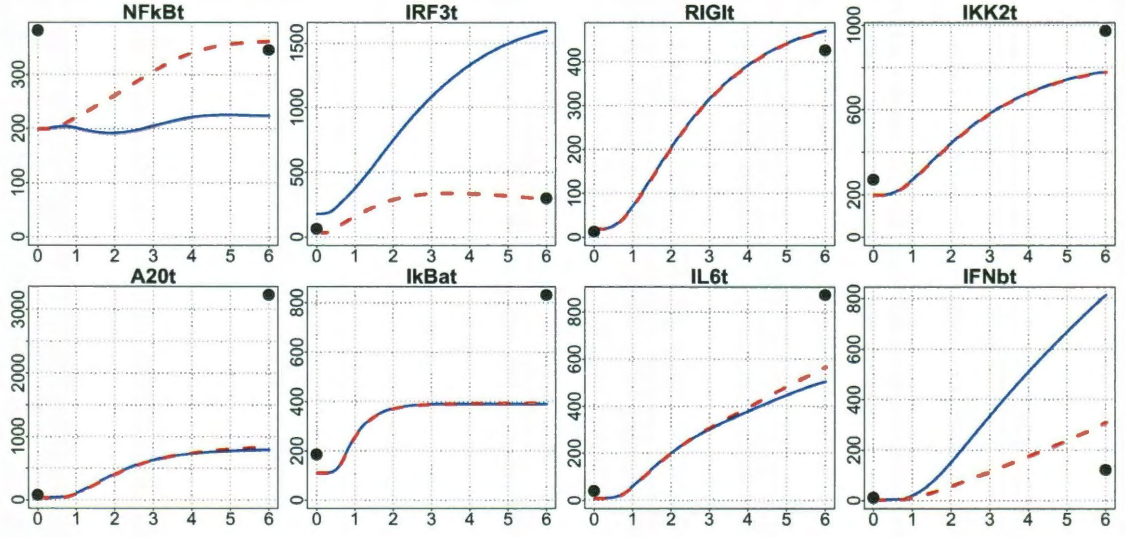


Figure C.2 : Simulation results: IRF3 Knock-down, compared to experimental results.

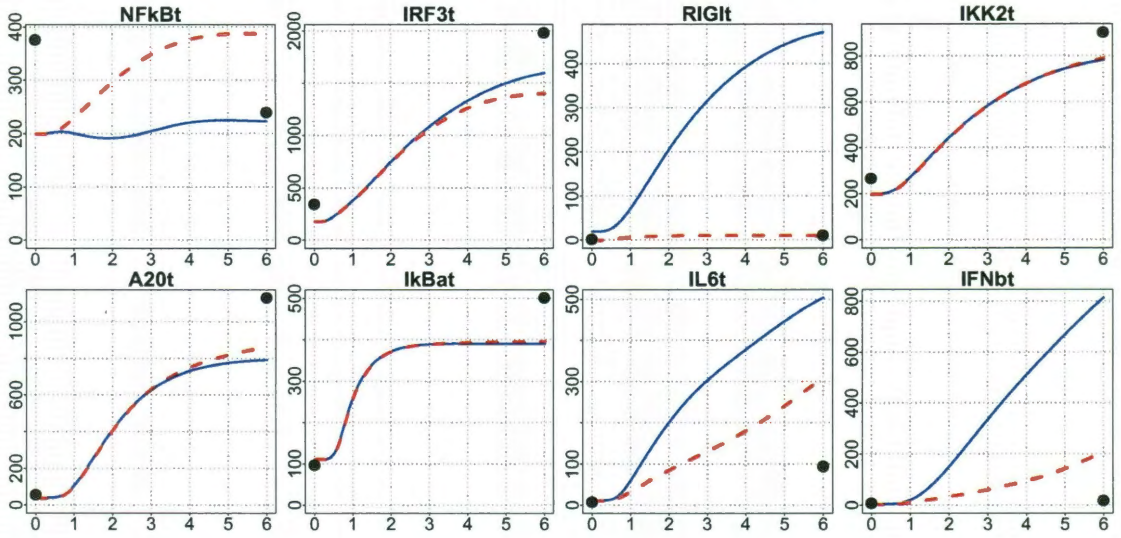


Figure C.3 : Simulation results: RIGI Knock-down, compared to experimental results.

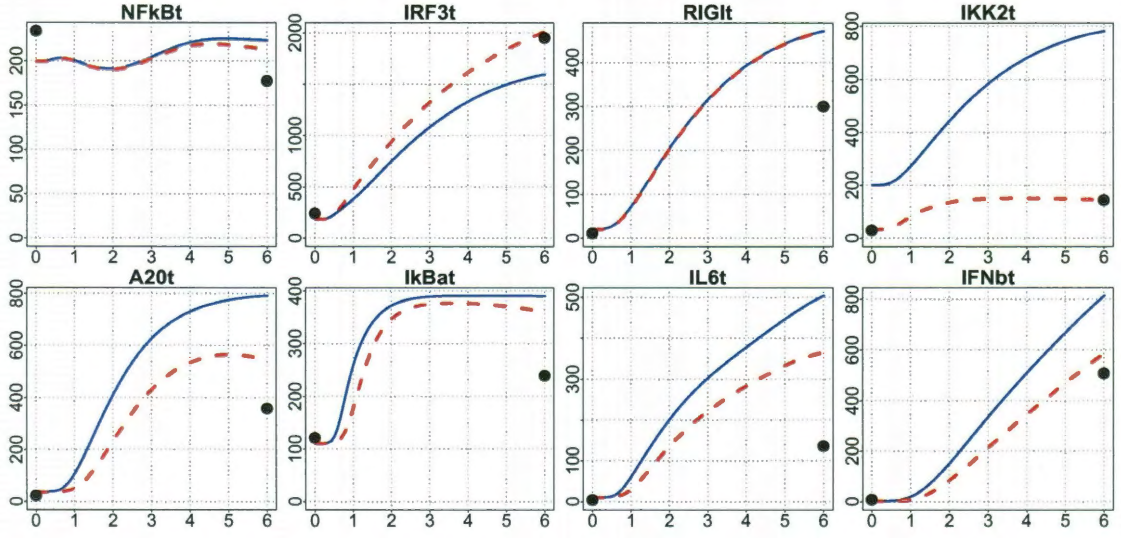


Figure C.4 : Simulation results: IKK2 Knock-down, compared to experimental results.

- Internal standard plasmids containing the same PCR sequence were used to determine the number of transcript copies per cell.
- Serial dilutions of plasmid were assayed in Q-PCR. 5 μL of the standard was used in each reaction. Molecular weight of plasmid is given as $2 * 320 gm/mole * 3973 bp = 2.542^6 gm/mol$.
- Ct is linearly related to the log of the standard (in number of molecules).
- Least squares regression was determined for Ct measurements of standard curve.

C.2 Complete set of coupled chemical equations that specifies the model

C.2.1 dsRNA, RIGI, MAVS, and TRAF3 interaction

C.2.1.1 RIGI ubiquitylation

$$\text{RIGI} + \text{dsRNA} \xrightarrow{\text{dsRNA\_recogn\_r / prot\_avg}} \text{RIGI}_{ub} \tag{C.2}$$

C.2.1.2 Ubiquitylated RIGI degradation

$$\text{RIGI}_{ub} \xrightarrow{\text{prot\_phosph\_degr\_r}} \otimes \tag{C.3}$$

C.2.1.3 RIGI-MAVS complex formation

$$\text{RIGI}_{ub} + \text{MAVS} \xrightarrow{\text{5e-1 * prot\_bind\_r / prot\_avg}} \text{RIGI-MAVS} \tag{C.4}$$

C.2.1.4 RIGI-MAVS degradation

$$\text{RIGI-MAVS} \xrightarrow{\text{prot\_phosph\_degr\_r}} \otimes \tag{C.5}$$

C.2.1.5 RIGI-MAVS-TRAF3 complex formation

$$\text{RIGI-MAVS} + \text{TRAF3} \xrightarrow{\text{5e-1 * prot\_bind\_r / prot\_avg}} \text{RIGI-MAVS-TRAF3} \tag{C.6}$$

C.2.1.6 RIGI-MAVS-TRAF3 degradation

$$\text{RIGI-MAVS-TRAF3} \xrightarrow{\text{prot\_phosph\_degr\_r}} \otimes \tag{C.7}$$

C.2.2 IKK1, RIGI-MAVS-TRAF3, and A20 interaction

C.2.2.1 IKK1 activation

$$\text{RIGI-MAVS-TRAF3} + \text{IKK1}_i \xrightarrow{\text{6e-1 * prot\_activ\_r / prot\_avg}} \text{RIGI-MAVS-TRAF3} + \text{IKK1}_a \tag{C.8}$$

C.2.2.2 Active IKK1 degradation

$$\text{IKK1}_a \xrightarrow{\text{1e1 * prot\_phosph\_degr\_r}} \otimes \tag{C.9}$$

C.2.2.3 Active IKK1 degradation induced by A20

$$\text{IKK1}_a + \text{A20} \xrightarrow{\text{prot\_phosph\_degr\_r / prot\_avg}} \text{A20} \tag{C.10}$$

C.2.3 IRF3 and IKK1 interaction

C.2.3.1 IRF3 activation

$$\text{IKK1}_a + \text{IRF3}_i \xrightarrow{\text{5 * prot\_activ\_r / prot\_avg}} \text{IKK1}_a + \text{IRF3}_a \tag{C.11}$$

C.2.3.2 Active IRF3 degradation

$$\text{IRF3}_a \xrightarrow{\text{3 * prot\_phosph\_degr\_r}} \otimes \tag{C.12}$$

C.2.3.3 Active IRF3 nuclear import

$$\text{IRF3}_a \xrightarrow{\text{prot\_import\_r}} \text{IRF3}_{an} \tag{C.13}$$

C.2.3.4 Nuclear active IRF3 degradation

$$\text{IRF3}_{an} \xrightarrow{\text{3 * prot\_phosph\_degr\_r}} \otimes \tag{C.14}$$

C.2.4 NFIL6 and IKK1 interaction

C.2.4.1 NFIL6 activation

$$\mathrm{IKK1}_a + \mathrm{NFIL6}_i \xrightarrow{5 * \text{prot\_activ\_r / prot\_avg}} \mathrm{IKK1}_a + \mathrm{NFIL6}_a \tag{C.15}$$

C.2.4.2 Active NFIL6 nuclear import

$$\mathrm{NFIL6}_a \xrightarrow{\text{prot\_import\_r}} \mathrm{NFIL6}_{an} \tag{C.16}$$

C.2.4.3 Nuclear active NFIL6 degradation

$$\mathrm{NFIL6}_{an} \xrightarrow{\text{prot\_phosph\_degr\_r}} \otimes \tag{C.17}$$

C.2.5 dsRNA, TLR3, TRIF, and TRAF6 interaction

C.2.5.1 TLR3 activation

$$\mathrm{TLR3}_i + \mathrm{dsRNA} \xrightarrow{\text{dsRNA\_recogn\_r / prot\_avg}} \mathrm{TLR3}_a \tag{C.18}$$

C.2.5.2 Active TLR3 degradation

$$\mathrm{TLR3}_a \xrightarrow{\text{prot\_phosph\_degr\_r}} \otimes \tag{C.19}$$

C.2.5.3 TLR3-TRIF complex formation

$$\mathrm{TLR3}_a + \mathrm{TRIF} \xrightarrow{\text{prot\_bind\_r / prot\_avg}} \mathrm{TLR3\text{-}TRIF} \tag{C.20}$$

C.2.5.4 TLR3-TRIF degradation

$$\mathrm{TLR3\text{-}TRIF} \xrightarrow{\text{prot\_phosph\_degr\_r}} \otimes \tag{C.21}$$

C.2.5.5 TLR3-TRIF-TRAF6 complex formation

$$\mathrm{TLR3\text{-}TRIF} + \mathrm{TRAF6} \xrightarrow{\text{prot\_bind\_r / prot\_avg}} \mathrm{TLR3\text{-}TRIF\text{-}TRAF6} \tag{C.22}$$

C.2.5.6 TLR3-TRIF-TRAF6 degradation

$$\text{TLR3-TRIF-TRAF6} \xrightarrow{\text{prot\_phosph\_degr\_r}} \otimes \tag{C.23}$$

C.2.6 IKK2, TLR3-TRIF-TRAF6, and A20 interaction

C.2.6.1 IKK2 activation

$$\text{TLR3-TRIF-TRAF6} + \text{IKK2}_i \xrightarrow{\text{prot\_activ\_r / prot\_avg}} \text{TLR3-TRIF-TRAF6} + \text{IKK2}_a \tag{C.24}$$

C.2.6.2 Active IKK2 degradation

$$\text{IKK2}_a \xrightarrow{\text{prot\_phosph\_degr\_r}} \otimes \tag{C.25}$$

C.2.6.3 Active IKK2 degradation induced by A20

$$\text{IKK2}_a + \text{A20} \xrightarrow{\text{prot\_phosph\_degr\_r / prot\_avg}} \text{A20} \tag{C.26}$$

C.2.7 NFκB, IκBα, and IKK2 interaction

C.2.7.1 IκBα phosphorylation

$$\text{IKK2}_a + \text{I}\kappa\text{B}\alpha \xrightarrow{\text{1e1 * prot\_activ\_r / prot\_avg}} \text{IKK2}_a + \text{I}\kappa\text{B}\alpha_p \tag{C.27}$$

C.2.7.2 Phosphorylated IκBα degradation

$$\text{I}\kappa\text{B}\alpha_p \xrightarrow{\text{1e1 * prot\_phosph\_degr\_r}} \otimes \tag{C.28}$$

C.2.7.3 NFκB-IκBα degradation

$$\text{NF}\kappa\text{B-I}\kappa\text{B}\alpha \xrightarrow{\text{prot\_degr\_r}} \otimes \tag{C.29}$$

C.2.7.4 NFκB-IκBα phosphorylation

$$\text{IKK2}_a + \text{NF}\kappa\text{B-I}\kappa\text{B}\alpha \xrightarrow{\text{1e1 * prot\_activ\_r / prot\_avg}} \text{IKK2}_a + \text{NF}\kappa\text{B-I}\kappa\text{B}\alpha_p \tag{C.30}$$

C.2.7.5 Phosphorylated NFκB-IκBα dissociation and degradation of IκBα

$$\text{NF}\kappa\text{B-I}\kappa\text{B}\alpha_p \xrightarrow{\text{1e1 * prot\_phosph\_degr\_r}} \text{NF}\kappa\text{B} \tag{C.31}$$

C.2.7.6 NFκB and IκBα binding

$$\text{NF}\kappa\text{B} + \text{I}\kappa\text{B}\alpha \xrightarrow{\text{1e1 * prot\_bind\_r / prot\_avg}} \text{NF}\kappa\text{B-I}\kappa\text{B}\alpha \tag{C.32}$$

C.2.7.7 NFκB-IκBα spontaneous dissociation

$$\text{NF}\kappa\text{B-I}\kappa\text{B}\alpha \xrightarrow{\text{prot\_diss\_r}} \text{NF}\kappa\text{B} + \text{I}\kappa\text{B}\alpha \tag{C.33}$$

C.2.7.8 NFκB nuclear import

$$\text{NF}\kappa\text{B} \xrightarrow{\text{prot\_import\_r}} \text{NF}\kappa\text{B}_n \tag{C.34}$$

IκBα nuclear import and export are treated in C.2.9, equations C.50 and C.52.

C.2.7.9 Nuclear NFκB and IκBα binding

$$\text{NF}\kappa\text{B}_n + \text{I}\kappa\text{B}\alpha_n \xrightarrow{\text{5 * 1e1 * prot\_bind\_r / prot\_avg}} \text{NF}\kappa\text{B}_n\text{-I}\kappa\text{B}\alpha_n \tag{C.35}$$

C.2.7.10 Nuclear NFκB-IκBα degradation

$$\text{NF}\kappa\text{B}_n\text{-I}\kappa\text{B}\alpha_n \xrightarrow{\text{prot\_degr\_r}} \otimes \tag{C.36}$$

C.2.7.11 NFκB-IκBα nuclear export

$$\text{NF}\kappa\text{B}_n\text{-I}\kappa\text{B}\alpha_n \xrightarrow{\text{prot\_export\_r}} \text{NF}\kappa\text{B-I}\kappa\text{B}\alpha \tag{C.37}$$

C.2.8 A20 regulation

C.2.8.1 A20 gene activation by NFκB$_n$

$$\text{NF}\kappa\text{B}_n + \text{A20}_g \xrightarrow{\text{1e-1 * gene\_act\_r / prot\_avg}} \text{A20}_g\text{-NF}\kappa\text{B}_n \tag{C.38}$$

C.2.8.2 A20 gene inactivation by IκBα_n

$$\text{I}\kappa\text{B}\alpha_n + \text{A20}_g\text{-NF}\kappa\text{B}_n \xrightarrow{\text{gene\_inact\_r / prot\_avg}} \text{A20}_g + \text{NF}\kappa\text{B}_n\text{-I}\kappa\text{B}\alpha_n \tag{C.39}$$

C.2.8.3 A20 mRNA transcription

$$\text{A20}_g\text{-NF}\kappa\text{B}_n \xrightarrow{\text{mRNA\_transc\_r}} \text{A20}_g\text{-NF}\kappa\text{B}_n + \text{A20}_t \tag{C.40}$$

C.2.8.4 A20 transcript degradation

$$\text{A20}_t \xrightarrow{\text{2 * mRNA\_degr\_r}} \otimes \tag{C.41}$$

C.2.8.5 A20 protein translation

$$\text{A20}_t \xrightarrow{\text{prot\_transl\_r}} \text{A20}_t + \text{A20} \tag{C.42}$$

C.2.8.6 A20 protein degradation

$$\text{A20} \xrightarrow{\text{prot\_degr\_r}} \otimes \tag{C.43}$$

C.2.9 IκBα regulation

C.2.9.1 IκBα gene activation by NFκB$_n$

$$\text{NF}\kappa\text{B}_n + \text{I}\kappa\text{B}\alpha_g \xrightarrow{\text{1e-1 * gene\_act\_r / prot\_avg}} \text{I}\kappa\text{B}\alpha_g\text{-NF}\kappa\text{B}_n \tag{C.44}$$

C.2.9.2 IκBα gene inactivation by IκBα_n

$$\text{I}\kappa\text{B}\alpha_n + \text{I}\kappa\text{B}\alpha_g\text{-NF}\kappa\text{B}_n \xrightarrow{\text{gene\_inact\_r / prot\_avg}} \text{I}\kappa\text{B}\alpha_g + \text{NF}\kappa\text{B}_n\text{-I}\kappa\text{B}\alpha_n \tag{C.45}$$

C.2.9.3 IκBα mRNA transcription

$$\text{I}\kappa\text{B}\alpha_g\text{-NF}\kappa\text{B}_n \xrightarrow{\text{mRNA\_transc\_r}} \text{I}\kappa\text{B}\alpha_g\text{-NF}\kappa\text{B}_n + \text{I}\kappa\text{B}\alpha_t \tag{C.46}$$

C.2.9.4 IκBα transcript degradation

$$\text{I}\kappa\text{B}\alpha_t \xrightarrow{\text{5 * mRNA\_degr\_r}} \otimes \tag{C.47}$$

C.2.9.5 IκBα protein translation

$$\text{I}\kappa\text{B}\alpha_t \xrightarrow{\text{prot\_transl\_r}} \text{I}\kappa\text{B}\alpha_t + \text{I}\kappa\text{B}\alpha \tag{C.48}$$

C.2.9.6 IκBα protein degradation

$$\text{I}\kappa\text{B}\alpha \xrightarrow{\text{prot\_degr\_r}} \otimes \tag{C.49}$$

C.2.9.7 IκBα nuclear import

$$\text{I}\kappa\text{B}\alpha \xrightarrow{\text{prot\_import\_r}} \text{I}\kappa\text{B}\alpha_n \tag{C.50}$$

C.2.9.8 Nuclear IκBα protein degradation

$$\text{I}\kappa\text{B}\alpha_n \xrightarrow{\text{prot\_degr\_r}} \otimes \tag{C.51}$$

C.2.9.9 IκBα nuclear export

$$\text{I}\kappa\text{B}\alpha_n \xrightarrow{\text{5e-1 * prot\_export\_r}} \text{I}\kappa\text{B}\alpha \tag{C.52}$$

C.2.10 IFNβ regulation

C.2.10.1 IFNβ gene partial activation by NFκB$_n$

$$\text{NF}\kappa\text{B}_n + \text{IFN}\beta_g \xrightarrow{\text{1e-1 * gene\_act\_r / prot\_avg}} \text{IFN}\beta_g\text{-NF}\kappa\text{B}_n \tag{C.53}$$

C.2.10.2 Partially active IFNβ gene inactivation by IκBα_n

$$\text{I}\kappa\text{B}\alpha_n + \text{IFN}\beta_g\text{-NF}\kappa\text{B}_n \xrightarrow{\text{gene\_inact\_r / prot\_avg}} \text{IFN}\beta_g + \text{NF}\kappa\text{B}_n\text{-I}\kappa\text{B}\alpha_n \tag{C.54}$$

C.2.10.3 Partially active IFNβ gene spontaneous inactivation

$$\text{IFN}\beta_g\text{-NF}\kappa\text{B}_n \xrightarrow{\text{3e-1 * gene\_inact\_r}} \text{IFN}\beta_g + \text{NF}\kappa\text{B}_n \tag{C.55}$$

C.2.10.4 IFNβ gene partial activation by IRF3$_{an}$

$$\text{IRF3}_{an} + \text{IFN}\beta_g \xrightarrow{\text{1e-1 * gene\_act\_r / prot\_avg}} \text{IFN}\beta_g\text{-IRF3}_{an} \tag{C.56}$$

C.2.10.5 Partially active IFNβ gene spontaneous inactivation

$$\text{IFN}\beta_g\text{-IRF3}_{an} \xrightarrow{\text{5e-2 * gene\_inact\_r}} \text{IFN}\beta_g + \text{IRF3}_{an} \tag{C.57}$$

C.2.10.6 IFNβ gene full activation by IRF3$_{an}$

$$\text{IRF3}_{an} + \text{IFN}\beta_g\text{-NF}\kappa\text{B}_n \xrightarrow{\text{gene\_act\_r / prot\_avg}} \text{IFN}\beta_g\text{-NF}\kappa\text{B}_n\text{-IRF3}_{an} \tag{C.58}$$

C.2.10.7 IFNβ gene full activation by NFκB$_n$

$$\text{NF}\kappa\text{B}_n + \text{IFN}\beta_g\text{-IRF3}_{an} \xrightarrow{\text{1e-1 * gene\_act\_r / prot\_avg}} \text{IFN}\beta_g\text{-NF}\kappa\text{B}_n\text{-IRF3}_{an} \tag{C.59}$$

C.2.10.8 Fully active IFNβ gene partial inactivation by IκBα_n

$$\text{I}\kappa\text{B}\alpha_n\text{+IFN}\beta_g\text{-NF}\kappa\text{B}_n\text{-IRF3}_{an} \xrightarrow{\text{1e-2 * gene\_inact\_r / prot\_avg}} \text{IFN}\beta_g\text{-IRF3}_{an}\text{+NF}\kappa\text{B}_n\text{-I}\kappa\text{B}\alpha_n \tag{C.60}$$

C.2.10.9 Fully active IFNβ gene spontaneous partial inactivation 1

$$\text{IFN}\beta_g\text{-NF}\kappa\text{B}_n\text{-IRF3}_{an} \xrightarrow{\text{1e-3 * gene\_inact\_r}} \text{IFN}\beta_g\text{-IRF3}_{an} + \text{NF}\kappa\text{B}_n \tag{C.61}$$

C.2.10.10 Fully active IFNβ gene spontaneous partial inactivation 2

$$\text{IFN}\beta_g\text{-NF}\kappa\text{B}_n\text{-IRF3}_{an} \xrightarrow{\text{1e-3 * gene\_inact\_r}} \text{IFN}\beta_g\text{-NF}\kappa\text{B}_n + \text{IRF3}_{an} \tag{C.62}$$

C.2.10.11 IFNβ mRNA transcription by fully activated gene

$$\text{IFN}\beta_g\text{-NF}\kappa\text{B}_n\text{-IRF3}_{an} \xrightarrow{\text{mRNA\_transc\_r}} \text{IFN}\beta_g\text{-NF}\kappa\text{B}_n\text{-IRF3}_{an} + \text{IFN}\beta_t \tag{C.63}$$

C.2.10.12 IFNβ mRNA transcription by gene partially activated by NFκB$_n$

$$\text{IFN}\beta_g\text{-NF}\kappa\text{B}_n \xrightarrow{\text{3e-1 * mRNA\_transc\_r}} \text{IFN}\beta_g\text{-NF}\kappa\text{B}_n + \text{IFN}\beta_t \tag{C.64}$$

C.2.10.13 IFNβ mRNA transcription by gene partially activated by IRF3$_{an}$

$$\text{IFN}\beta_g\text{-IRF3}_{an} \xrightarrow{\text{3e-1 * mRNA\_transc\_r}} \text{IFN}\beta_g\text{-IRF3}_{an} + \text{IFN}\beta_t \tag{C.65}$$

C.2.10.14 IFNβ transcript degradation

$$\text{IFN}\beta_t \xrightarrow{\text{mRNA\_degr\_r}} \otimes \tag{C.66}$$

C.2.10.15 IFNβ cytokine translation

$$\text{IFN}\beta_t \xrightarrow{\text{prot\_transl\_r}} \text{IFN}\beta_t + \text{IFN}\beta \tag{C.67}$$

C.2.10.16 IFNβ cytokine degradation

$$\text{IFN}\beta \xrightarrow{\text{prot\_degr\_r}} \otimes \tag{C.68}$$

C.2.11 IL6 regulation

C.2.11.1 IL6 gene partial activation by NFκB$_n$

$$\text{NF}\kappa\text{B}_n + \text{IL6}_g \xrightarrow{\text{gene\_act\_r / prot\_avg}} \text{IL6}_g\text{-NF}\kappa\text{B}_n \tag{C.69}$$

C.2.11.2 Partially active IL6 gene inactivation by IκBα_n

$$\text{I}\kappa\text{B}\alpha_n + \text{IL6}_g\text{-NF}\kappa\text{B}_n \xrightarrow{\text{gene\_inact\_r / prot\_avg}} \text{IL6}_g + \text{NF}\kappa\text{B}_n\text{-I}\kappa\text{B}\alpha_n \tag{C.70}$$

C.2.11.3 Partially active IL6 gene spontaneous inactivation 1

$$\text{IL6}_g\text{-NF}\kappa\text{B}_n \xrightarrow{\text{gene\_inact\_r}} \text{IL6}_g + \text{NF}\kappa\text{B}_n \tag{C.71}$$

C.2.11.4 IL6 gene partial activation by NFIL6$_{an}$

$$\text{NFIL6}_{an} + \text{IL6}_g \xrightarrow{\text{1e1 * gene\_act\_r / prot\_avg}} \text{IL6}_g\text{-NFIL6}_{an} \tag{C.72}$$

C.2.11.5 Partially active IL6 gene spontaneous inactivation 2

$$\text{IL6}_g\text{-NFIL6}_{an} \xrightarrow{\text{gene\_inact\_r}} \text{IL6}_g + \text{NFIL6}_{an} \tag{C.73}$$

C.2.11.6 IL6 gene full activation by NFIL6$_{an}$

$$\text{NFIL6}_{an} + \text{IL6}_g\text{-NF}\kappa\text{B}_n \xrightarrow{\text{gene\_act\_r / prot\_avg}} \text{IL6}_g\text{-NF}\kappa\text{B}_n\text{-NFIL6}_{an} \tag{C.74}$$

C.2.11.7 IL6 gene full activation by $NF\kappa B_n$

$$NF\kappa B_n + IL6_g\text{-}NFIL6_{an} \xrightarrow{\text{gene\_act\_r / prot\_avg}} IL6_g\text{-}NF\kappa B_n\text{-}NFIL6_{an} \tag{C.75}$$

C.2.11.8 Fully active IL6 gene partial inactivation by $I\kappa B\alpha_n$

$$I\kappa B\alpha_n + IL6_g\text{-}NF\kappa B_n\text{-}NFIL6_{an} \xrightarrow{\text{gene\_inact\_r / prot\_avg}} IL6_g\text{-}NFIL6_{an} + NF\kappa B_n\text{-}I\kappa B\alpha_n \tag{C.76}$$

C.2.11.9 Fully active IL6 gene spontaneous partial inactivation 2

$$IL6_g\text{-}NF\kappa B_n\text{-}NFIL6_{an} \xrightarrow{\text{gene\_inact\_r}} IL6_g\text{-}NFIL6_{an} + NF\kappa B_n \tag{C.77}$$

C.2.11.10 Fully active IL6 gene spontaneous partial inactivation 2

$$IL6_g\text{-}NF\kappa B_n\text{-}NFIL6_{an} \xrightarrow{\text{gene\_inact\_r}} IL6_g\text{-}NF\kappa B_n + NFIL6_{an} \tag{C.78}$$

C.2.11.11 IL6 mRNA transcription by fully activated gene

$$IL6_g\text{-}NF\kappa B_n\text{-}NFIL6_{an} \xrightarrow{\text{mRNA\_transc\_r}} IL6_g\text{-}NF\kappa B_n\text{-}NFIL6_{an} + IL6_t \tag{C.79}$$

C.2.11.12 IL6 mRNA transcription by gene partially activated by $NF\kappa B_n$

$$IL6_g\text{-}NF\kappa B_n \xrightarrow{\text{3e-1 * mRNA\_transc\_r}} IL6_g\text{-}NF\kappa B_n + IL6_t \tag{C.80}$$

C.2.11.13 IL6 mRNA transcription by gene partially activated by $NFIL6_{an}$

$$IL6_g\text{-}NFIL6_{an} \xrightarrow{\text{3e-1 * mRNA\_transc\_r}} IL6_g\text{-}NFIL6_{an} + IL6_t \tag{C.81}$$

C.2.11.14 IL6 transcript degradation

$$IL6_t \xrightarrow{\text{mRNA\_degr\_r}} \otimes \tag{C.82}$$

C.2.11.15 IL6 protein translation

$$IL6_t \xrightarrow{\text{prot\_transl\_r}} IL6_t + IL6 \tag{C.83}$$

C.2.11.16 IL6 protein degradation

$$\mathrm{IL6} \xrightarrow{\text{prot\_degr\_r}} \otimes \tag{C.84}$$

C.2.12 Inactive TLR3 regulation

C.2.12.1 TLR3 mRNA basal transcription

$$\mathrm{TLR3}_g \xrightarrow{\text{mRNA\_basal\_transc\_r}} \mathrm{TLR3}_g + \mathrm{TLR3}_t \tag{C.85}$$

C.2.12.2 TLR3 transcript degradation

$$\mathrm{TLR3}_t \xrightarrow{\text{mRNA\_degr\_r}} \otimes \tag{C.86}$$

C.2.12.3 Inactive TLR3 protein translation

$$\mathrm{TLR3}_t \xrightarrow{\text{prot\_transl\_r}} \mathrm{TLR3}_t + \mathrm{TLR3}_i \tag{C.87}$$

C.2.12.4 Inactive TLR3 protein degradation

$$\mathrm{TLR3}_i \xrightarrow{\text{prot\_degr\_r}} \otimes \tag{C.88}$$

C.2.13 TRAF3 regulation

C.2.13.1 TRAF3 mRNA basal transcription

$$\mathrm{TRAF3}_g \xrightarrow{\text{1e-1 * mRNA\_basal\_transc\_r}} \mathrm{TRAF3}_g + \mathrm{TRAF3}_t \tag{C.89}$$

C.2.13.2 TRAF3 transcript degradation

$$\mathrm{TRAF3}_t \xrightarrow{\text{mRNA\_degr\_r}} \otimes \tag{C.90}$$

C.2.13.3 TRAF3 protein translation

$$\mathrm{TRAF3}_t \xrightarrow{\text{prot\_transl\_r}} \mathrm{TRAF3}_t + \mathrm{TRAF3} \tag{C.91}$$

C.2.13.4 TRAF3 protein degradation

$$\text{TRAF3} \xrightarrow{\text{1e-1 * prot\_degr\_r}} \otimes \tag{C.92}$$

C.2.14 TRAF6 regulation

C.2.14.1 TRAF6 mRNA basal transcription

$$\text{TRAF6}_g \xrightarrow{\text{mRNA\_basal\_transc\_r}} \text{TRAF6}_g + \text{TRAF6}_t \tag{C.93}$$

C.2.14.2 TRAF6 transcript degradation

$$\text{TRAF6}_t \xrightarrow{\text{mRNA\_degr\_r}} \otimes \tag{C.94}$$

C.2.14.3 TRAF6 protein translation

$$\text{TRAF6}_t \xrightarrow{\text{prot\_transl\_r}} \text{TRAF6}_t + \text{TRAF6} \tag{C.95}$$

C.2.14.4 TRAF6 protein degradation

$$\text{TRAF6} \xrightarrow{\text{prot\_degr\_r}} \otimes \tag{C.96}$$

C.2.15 TRIF regulation

C.2.15.1 TRIF mRNA basal transcription

$$\text{TRIF}_g \xrightarrow{\text{mRNA\_basal\_transc\_r}} \text{TRIF}_g + \text{TRIF}_t \tag{C.97}$$

C.2.15.2 TRIF transcript degradation

$$\text{TRIF}_t \xrightarrow{\text{mRNA\_degr\_r}} \otimes \tag{C.98}$$

C.2.15.3 TRIF protein translation

$$\text{TRIF}_t \xrightarrow{\text{prot\_transl\_r}} \text{TRIF}_t + \text{TRIF} \tag{C.99}$$

C.2.15.4 TRIF protein degradation

$$\text{TRIF} \xrightarrow{\text{prot\_degr\_r}} \otimes \tag{C.100}$$

C.2.16 TLR3dTF regulation

C.2.16.1 TLR3dTF mRNA basal transcription

$$\text{TLR3dTF}_g \xrightarrow{\text{mRNA\_basal\_transc\_r}} \text{TLR3dTF}_g + \text{TLR3dTF}_t \tag{C.101}$$

C.2.16.2 TLR3dTF transcript degradation

$$\text{TLR3dTF}_t \xrightarrow{\text{mRNA\_degr\_r}} \otimes \tag{C.102}$$

C.2.16.3 Inactive TLR3dTF protein translation

$$\text{TLR3dTF}_t \xrightarrow{\text{prot\_transl\_r}} \text{TLR3dTF}_t + \text{TLR3dTF}_i \tag{C.103}$$

C.2.16.4 Inactive TLR3dTF protein degradation

$$\text{TLR3dTF}_i \xrightarrow{\text{prot\_degr\_r}} \otimes \tag{C.104}$$

C.2.16.5 TLR3dTF activation

$$\text{TLR3-TRIF-TRAF6+TLR3dTF}_i \xrightarrow{\text{prot\_activ\_r / prot\_avg}} \text{TLR3-TRIF-TRAF6+TLR3dTF}_a \tag{C.105}$$

C.2.16.6 Active TLR3dTF nuclear import

$$\text{TLR3dTF}_a \xrightarrow{\text{prot\_import\_r}} \text{TLR3dTF}_{an} \tag{C.106}$$

C.2.16.7 Nuclear active TLR3dTF degradation

$$\text{TLR3dTF}_{an} \xrightarrow{\text{3e-1 * prot\_phosph\_degr\_r}} \otimes \tag{C.107}$$

C.2.17 RIGI regulation

C.2.17.1 RIGI gene activation by TLR3dTF

$$\text{TLR3dTF}_{an} + \text{RIGI}_g \xrightarrow{\text{6e-1 * gene\_act\_r / prot\_avg}} \text{RIGI}_g\text{-TLR3dTF}_{an} \tag{C.108}$$

C.2.17.2 RIGI gene inactivation

$$\text{RIGI}_g\text{-TLR3dTF}_{an} \xrightarrow{\text{gene\_inact\_r}} \text{RIGI}_g + \text{TLR3dTF}_{an} \tag{C.109}$$

C.2.17.3 RIGI mRNA transcription

$$\text{RIGI}_g\text{-TLR3dTF}_{an} \xrightarrow{\text{mRNA\_transc\_r}} \text{RIGI}_g\text{-TLR3dTF}_{an} + \text{RIGI}_t \tag{C.110}$$

C.2.17.4 RIGI mRNA basal transcription

$$\text{RIGI}_g \xrightarrow{\text{1e-1 * mRNA\_basal\_transc\_r}} \text{RIGI}_g + \text{RIGI}_t \tag{C.111}$$

C.2.17.5 RIGI transcript degradation

$$\text{RIGI}_t \xrightarrow{\text{mRNA\_degr\_r}} \otimes \tag{C.112}$$

C.2.17.6 RIGI protein translation

$$\text{RIGI}_t \xrightarrow{\text{prot\_transl\_r}} \text{RIGI}_t + \text{RIGI} \tag{C.113}$$

C.2.17.7 RIGI protein degradation

$$\text{RIGI} \xrightarrow{\text{1e-1 * prot\_degr\_r}} \otimes \tag{C.114}$$

C.2.18 Inactive IKK1 regulation

C.2.18.1 IKK1 gene activation by TLR3dTF

$$\text{TLR3dTF}_{an} + \text{IKK1}_g \xrightarrow{\text{gene\_act\_r / prot\_avg}} \text{IKK1}_g\text{-TLR3dTF}_{an} \tag{C.115}$$

C.2.18.2 IKK1 gene inactivation

$$\text{IKK1}_g\text{-TLR3dTF}_{an} \xrightarrow{\text{gene\_inact\_r}} \text{IKK1}_g + \text{TLR3dTF}_{an} \tag{C.116}$$

C.2.18.3 IKK1 mRNA transcription

$$\text{IKK1}_g\text{-TLR3dTF}_{an} \xrightarrow{\text{6e-1 * mRNA\_transc\_r}} \text{IKK1}_g\text{-TLR3dTF}_{an} + \text{IKK1}_t \tag{C.117}$$

C.2.18.4 IKK1 mRNA basal transcription

$$\text{IKK1}_g \xrightarrow{\text{mRNA\_basal\_transc\_r}} \text{IKK1}_g + \text{IKK1}_t \tag{C.118}$$

C.2.18.5 IKK1 transcript degradation

$$\text{IKK1}_t \xrightarrow{\text{mRNA\_degr\_r}} \otimes \tag{C.119}$$

C.2.18.6 Inactive IKK1 protein translation

$$\text{IKK1}_t \xrightarrow{\text{prot\_transl\_r}} \text{IKK1}_t + \text{IKK1}_i \tag{C.120}$$

C.2.18.7 Inactive IKK1 protein degradation

$$\text{IKK1}_i \xrightarrow{\text{1e-1 * prot\_degr\_r}} \otimes \tag{C.121}$$

C.2.19 Inactive IKK2 regulation

C.2.19.1 IKK2 gene activation by TLR3dTF

$$\text{TLR3dTF}_{an} + \text{IKK2}_g \xrightarrow{\text{gene\_act\_r / prot\_avg}} \text{IKK2}_g\text{-TLR3dTF}_{an} \tag{C.122}$$

C.2.19.2 IKK2 gene inactivation

$$\text{IKK2}_g\text{-TLR3dTF}_{an} \xrightarrow{\text{gene\_inact\_r}} \text{IKK2}_g + \text{TLR3dTF}_{an} \tag{C.123}$$

C.2.19.3 IKK2 mRNA transcription

$$\text{IKK2}_g\text{-TLR3dTF}_{an} \xrightarrow{\text{mRNA\_transc\_r}} \text{IKK2}_g\text{-TLR3dTF}_{an} + \text{IKK2}_t \tag{C.124}$$

C.2.19.4 IKK2 mRNA basal transcription

$$\text{IKK2}_g \xrightarrow{\text{mRNA\_basal\_transc\_r}} \text{IKK2}_g + \text{IKK2}_t \tag{C.125}$$

C.2.19.5 IKK2 transcript degradation

$$\text{IKK2}_t \xrightarrow{\text{mRNA\_degr\_r}} \otimes \tag{C.126}$$

C.2.19.6 Inactive IKK2 protein translation

$$\text{IKK2}_t \xrightarrow{\text{prot\_transl\_r}} \text{IKK2}_t + \text{IKK2}_i \tag{C.127}$$

C.2.19.7 Inactive IKK2 protein degradation

$$\text{IKK2}_i \xrightarrow{\text{prot\_degr\_r}} \otimes \tag{C.128}$$

C.2.20 MAVS regulation

C.2.20.1 MAVS gene activation by TLR3dTF

$$\text{TLR3dTF}_{an} + \text{MAVS}_g \xrightarrow{\text{gene\_act\_r / prot\_avg}} \text{MAVS}_g\text{-TLR3dTF}_{an} \tag{C.129}$$

C.2.20.2 MAVS gene inactivation

$$\text{MAVS}_g\text{-TLR3dTF}_{an} \xrightarrow{\text{gene\_inact\_r}} \text{MAVS}_g + \text{TLR3dTF}_{an} \tag{C.130}$$

C.2.20.3 MAVS mRNA transcription

$$\text{MAVS}_g\text{-TLR3dTF}_{an} \xrightarrow{\text{5e-1 * mRNA\_transc\_r}} \text{MAVS}_g\text{-TLR3dTF}_{an} + \text{MAVS}_t \tag{C.131}$$

C.2.20.4 MAVS mRNA basal transcription

$$\text{MAVS}_g \xrightarrow{\text{1e-1 * mRNA\_basal\_transc\_r}} \text{MAVS}_g + \text{MAVS}_t \tag{C.132}$$

C.2.20.5 MAVS transcript degradation

$$\text{MAVS}_t \xrightarrow{\text{mRNA\_degr\_r}} \otimes \tag{C.133}$$

C.2.20.6 MAVS protein translation

$$\text{MAVS}_t \xrightarrow{\text{prot\_transl\_r}} \text{MAVS}_t + \text{MAVS} \tag{C.134}$$

C.2.20.7 MAVS protein degradation

$$\text{MAVS} \xrightarrow{\text{1e-1 * prot\_degr\_r}} \otimes \tag{C.135}$$

C.2.21 Inactive IRF3 regulation

C.2.21.1 IRF3 gene inhibition by NFκB

$$\text{NF}\kappa\text{B}_n + \text{IRF3}_g \xrightarrow{\text{5 * gene\_act\_r / prot\_avg}} \text{IRF3}_g\text{-NF}\kappa\text{B}_n \tag{C.136}$$

C.2.21.2 IRF3 gene inhibition inactivation

$$\text{IRF3}_g\text{--NF}\kappa\text{B}_n \xrightarrow{\text{gene\_inact\_r}} \text{IRF3}_g + \text{-NF}\kappa\text{B}_n \tag{C.137}$$

C.2.21.3 IRF3 gene activation by TLR3dTF

$$\text{TLR3dTF}_{an} + \text{IRF3}_g \xrightarrow{\text{2 * gene\_act\_r / prot\_avg}} \text{IRF3}_g\text{-TLR3dTF}_{an} \tag{C.138}$$

C.2.21.4 IRF3 gene inactivation

$$\text{IRF3}_g\text{-TLR3dTF}_{an} \xrightarrow{\text{gene\_inact\_r}} \text{IRF3}_g + \text{TLR3dTF}_{an} \tag{C.139}$$

C.2.21.5 IRF3 mRNA transcription

$$\text{IRF3}_g\text{-TLR3dTF}_{an} \xrightarrow{\text{3 * mRNA\_transc\_r}} \text{IRF3}_g\text{-TLR3dTF}_{an} + \text{IRF3}_t \tag{C.140}$$

C.2.21.6 IRF3 mRNA basal transcription

$$\text{IRF3}_g \xrightarrow{\text{mRNA\_basal\_transc\_r}} \text{IRF3}_g + \text{IRF3}_t \tag{C.141}$$

C.2.21.7 IRF3 transcript degradation

$$\text{IRF3}_t \xrightarrow{\text{mRNA\_degr\_r}} \otimes \tag{C.142}$$

C.2.21.8 Double-inactive IRF3 protein translation

$$\mathrm{IRF3}_t \xrightarrow{\text{1e-1 * prot\_transl\_r}} \mathrm{IRF3}_t + \mathrm{IRF3}_{ii} \tag{C.143}$$

C.2.21.9 Double-inactive IRF3 protein degradation

$$\mathrm{IRF3}_{ii} \xrightarrow{\text{1e-1 * prot\_degr\_r}} \otimes \tag{C.144}$$

C.2.21.10 Maturation of double-inactive IRF3 protein

$$\mathrm{IRF3}_{ii} \xrightarrow{\text{1e-1 * prot\_bind\_r}} \mathrm{IRF3}_i \tag{C.145}$$

C.2.21.11 Inactive IRF3 protein degradation

$$\mathrm{IRF3}_i \xrightarrow{\text{1e-1 * prot\_degr\_r}} \otimes \tag{C.146}$$

C.2.21.12 Inactive IRF3 nuclear import

$$\mathrm{IRF3}_i \xrightarrow{\text{prot\_import\_r}} \mathrm{IRF3}_{in} \tag{C.147}$$

C.2.21.13 Inactive IRF3 nuclear export

$$\mathrm{IRF3}_{in} \xrightarrow{\text{1e1 * prot\_export\_r}} \mathrm{IRF3}_i \tag{C.148}$$

C.2.21.14 Nuclear inactive IRF3 protein degradation

$$\mathrm{IRF3}_{in} \xrightarrow{\text{1e-1 * prot\_degr\_r}} \otimes \tag{C.149}$$

C.2.22 Inactive NFIL6 regulation

C.2.22.1 NFIL6 gene activation by TLR3dTF

$$\mathrm{TLR3dTF}_{an} + \mathrm{NFIL6}_g \xrightarrow{\text{gene\_act\_r / prot\_avg}} \mathrm{NFIL6}_g\text{-}\mathrm{TLR3dTF}_{an} \tag{C.150}$$

C.2.22.2 NFIL6 gene inactivation

$$\mathrm{NFIL6}_g\text{-}\mathrm{TLR3dTF}_{an} \xrightarrow{\text{gene\_inact\_r}} \mathrm{NFIL6}_g + \mathrm{TLR3dTF}_{an} \tag{C.151}$$

C.2.22.3 NFIL6 mRNA transcription

$$\text{NFIL6}_g\text{-TLR3dTF}_{an} \xrightarrow{\text{mRNA\_transc\_r}} \text{NFIL6}_g\text{-TLR3dTF}_{an} + \text{NFIL6}_t \quad \text{(C.152)}$$

C.2.22.4 NFIL6 mRNA basal transcription

$$\text{NFIL6}_g \xrightarrow{\text{mRNA\_basal\_transc\_r}} \text{NFIL6}_g + \text{NFIL6}_t \quad \text{(C.153)}$$

C.2.22.5 NFIL6 transcript degradation

$$\text{NFIL6}_t \xrightarrow{\text{mRNA\_degr\_r}} \otimes \quad \text{(C.154)}$$

C.2.22.6 Inactive NFIL6 protein translation

$$\text{NFIL6}_t \xrightarrow{\text{prot\_transl\_r}} \text{NFIL6}_t + \text{NFIL6}_i \quad \text{(C.155)}$$

C.2.22.7 Inactive NFIL6 protein degradation

$$\text{NFIL6}_i \xrightarrow{\text{prot\_degr\_r}} \otimes \quad \text{(C.156)}$$

C.2.23 NFκB regulation

C.2.23.1 NFκB gene inhibition by IRF3

$$\text{IRF3}_{an} + \text{NF}\kappa\text{B}_g \xrightarrow{\text{5 * gene\_act\_r / prot\_avg}} \text{NF}\kappa\text{B}_g\text{-IRF3}_{an} \quad \text{(C.157)}$$

C.2.23.2 NFκB gene inhibition inactivation

$$\text{NF}\kappa\text{B}_g\text{-IRF3}_{an} \xrightarrow{\text{1e-1 * gene\_inact\_r}} \text{NF}\kappa\text{B}_g + \text{IRF3}_{an} \quad \text{(C.158)}$$

C.2.23.3 NFκB gene activation by TLR3dTF

$$\text{TLR3dTF}_{an} + \text{NF}\kappa\text{B}_g \xrightarrow{\text{gene\_act\_r / prot\_avg}} \text{NF}\kappa\text{B}_g\text{-TLR3dTF}_{an} \quad \text{(C.159)}$$

C.2.23.4 NFκB gene inactivation

$$\text{NF}\kappa\text{B}_g\text{-TLR3dTF}_{an} \xrightarrow{\text{gene\_inact\_r}} \text{NF}\kappa\text{B}_g + \text{TLR3dTF}_{an} \quad \text{(C.160)}$$

C.2.23.5 NFκB mRNA transcription

$$\text{NF}\kappa\text{B}_g\text{-TLR3dTF}_{an} \xrightarrow{\text{5e-1 * mRNA\_transc\_r}} \text{NF}\kappa\text{B}_g\text{-TLR3dTF}_{an} + \text{NF}\kappa\text{B}_t \quad \text{(C.161)}$$

C.2.23.6 NFκB mRNA basal transcription

$$\text{NF}\kappa\text{B}_g \xrightarrow{\text{mRNA\_basal\_transc\_r}} \text{NF}\kappa\text{B}_g + \text{NF}\kappa\text{B}_t \quad \text{(C.162)}$$

C.2.23.7 NFκB transcript degradation

$$\text{NF}\kappa\text{B}_t \xrightarrow{\text{mRNA\_degr\_r}} \otimes \quad \text{(C.163)}$$

C.2.23.8 Inactive NFκB protein translation

$$\text{NF}\kappa\text{B}_t \xrightarrow{\text{prot\_transl\_r}} \text{NF}\kappa\text{B}_t + \text{NF}\kappa\text{B}_i \quad \text{(C.164)}$$

C.2.23.9 Inactive NFκB protein degradation

$$\text{NF}\kappa\text{B}_i \xrightarrow{\text{prot\_degr\_r}} \otimes \quad \text{(C.165)}$$

C.2.23.10 Activation of NFκB protein

$$\text{NF}\kappa\text{B}_i \xrightarrow{\text{1e-1 * prot\_bind\_r}} \text{NF}\kappa\text{B}_i \quad \text{(C.166)}$$

C.2.23.11 NFκB protein degradation

$$\text{NF}\kappa\text{B} \xrightarrow{\text{prot\_degr\_r}} \otimes \quad \text{(C.167)}$$

C.2.23.12 Nuclear NFκB protein degradation

$$\text{NF}\kappa\text{B}_n \xrightarrow{\text{prot\_degr\_r}} \otimes \quad \text{(C.168)}$$

C.2.24 ISG56 regulation

C.2.24.1 ISG56 gene activation by IRF3

$$\text{IRF3}_{an} + \text{ISG56}_g \xrightarrow{\text{5 * gene\_act\_r / prot\_avg}} \text{ISG56}_g\text{-IRF3}_{an} \quad \text{(C.169)}$$

C.2.24.2 ISG56 gene inactivation

$$\text{ISG56}_g\text{-IRF3}_{an} \xrightarrow{\text{gene\_inact\_r}} \text{ISG56}_g + \text{IRF3}_{an} \tag{C.170}$$

C.2.24.3 ISG56 mRNA transcription

$$\text{ISG56}_g\text{-IRF3}_{an} \xrightarrow{\text{mRNA\_transc\_r}} \text{ISG56}_g\text{-IRF3}_{an} + \text{ISG56}_t \tag{C.171}$$

C.2.24.4 ISG56 mRNA basal transcription

$$\text{ISG56}_g \xrightarrow{\text{1e-2 * mRNA\_basal\_transc\_r}} \text{ISG56}_g + \text{ISG56}_t \tag{C.172}$$

C.2.24.5 ISG56 transcript degradation

$$\text{ISG56}_t \xrightarrow{\text{mRNA\_degr\_r}} \otimes \tag{C.173}$$

C.2.24.6 ISG56 protein translation

$$\text{ISG56}_t \xrightarrow{\text{prot\_transl\_r}} \text{ISG56}_t + \text{ISG56} \tag{C.174}$$

C.2.24.7 ISG56 protein degradation

$$\text{ISG56} \xrightarrow{\text{prot\_degr\_r}} \otimes \tag{C.175}$$

Appendix D

Additional material for Chapter 8

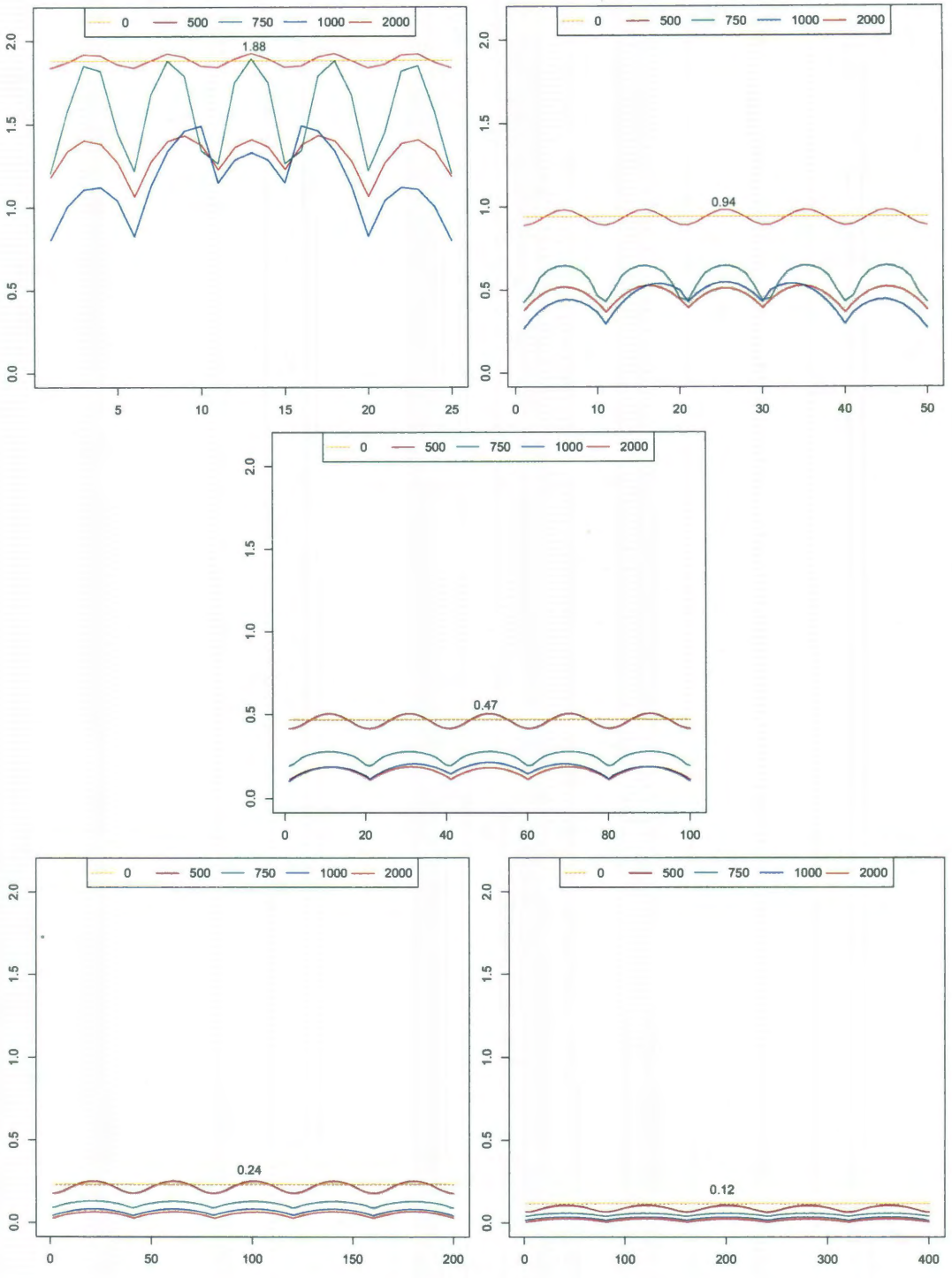


Figure D.1 : Cross sections of evolution of perturbed deterministic systems for different number of subdivisions. Free growth factor. Horizontal axis shows number of subdivisions of interval $0 - 1$.

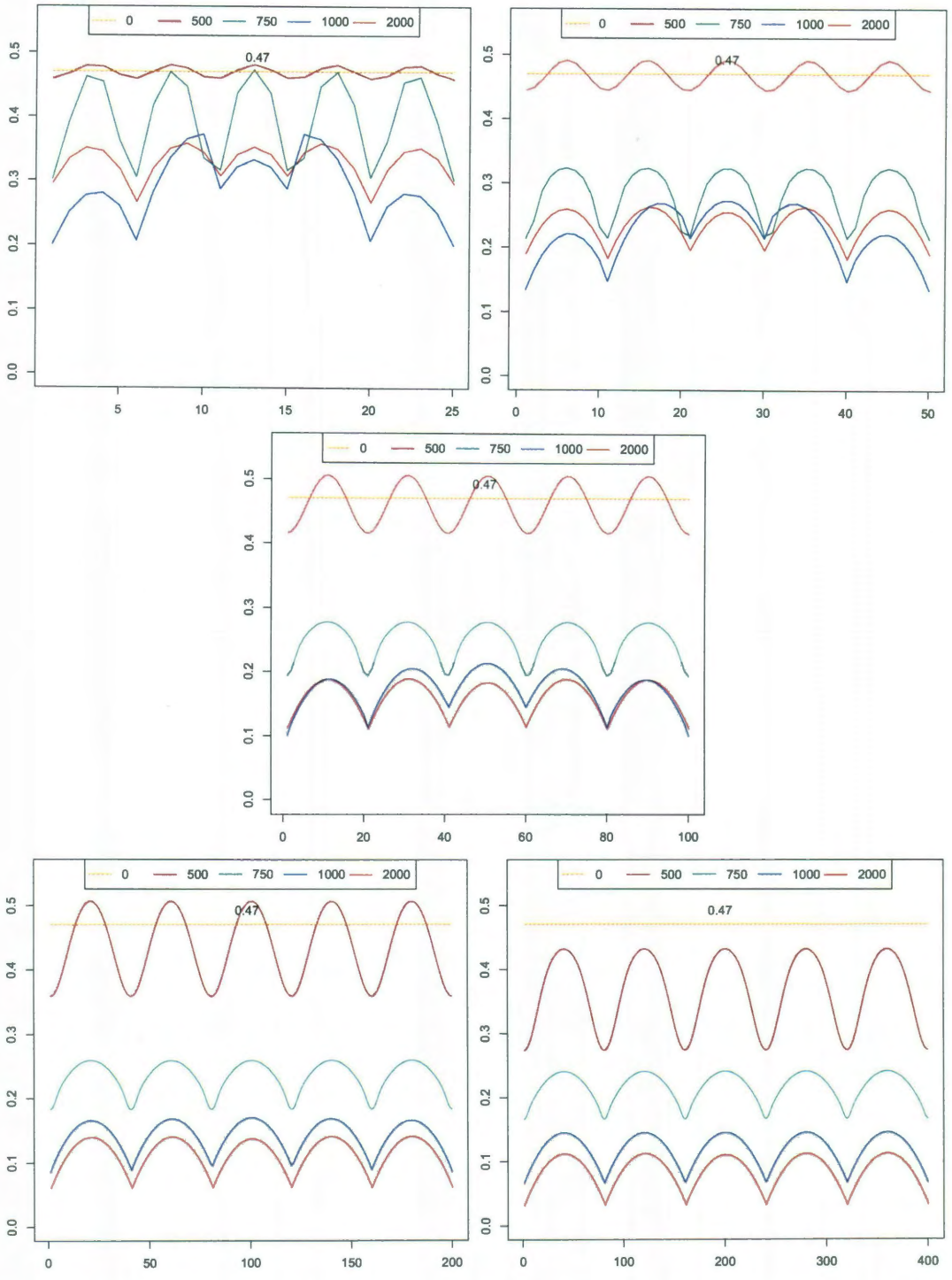


Figure D.2 : Cross sections of evolution of perturbed deterministic systems for different number of subdivisions, rescaled to show the same spatially homogeneous steady state as the main case (100 subdivisions). Free growth factor. Horizontal axis shows number of subdivisions of interval $0 - 1$.

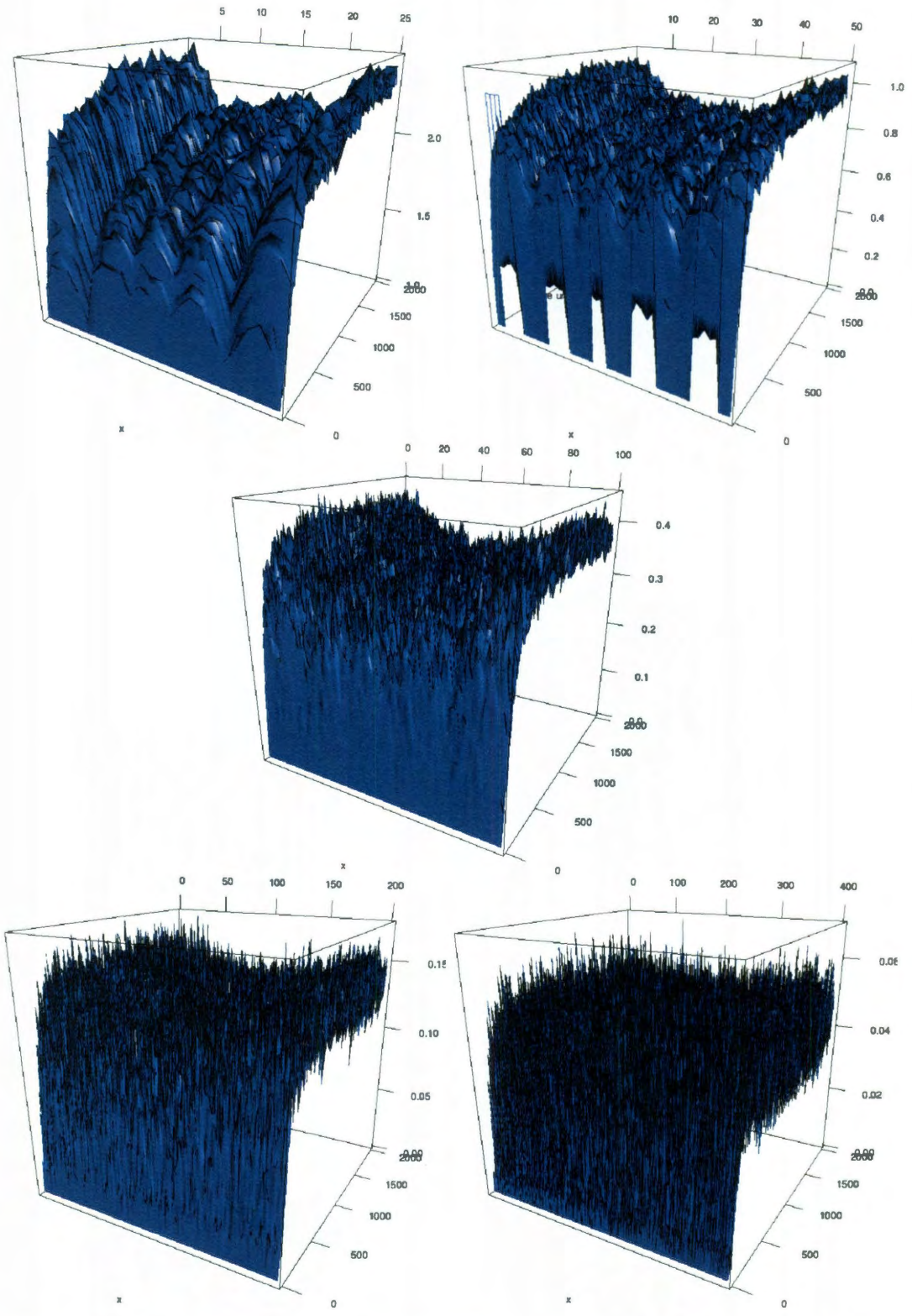


Figure D.3 : Stochastic reaction-diffusion system. Pre-cancerous cells. Averages of 1000 runs for different number of subdivisions. Initial conditions: end values of corresponding perturbed deterministic system rounded to nearest integer.

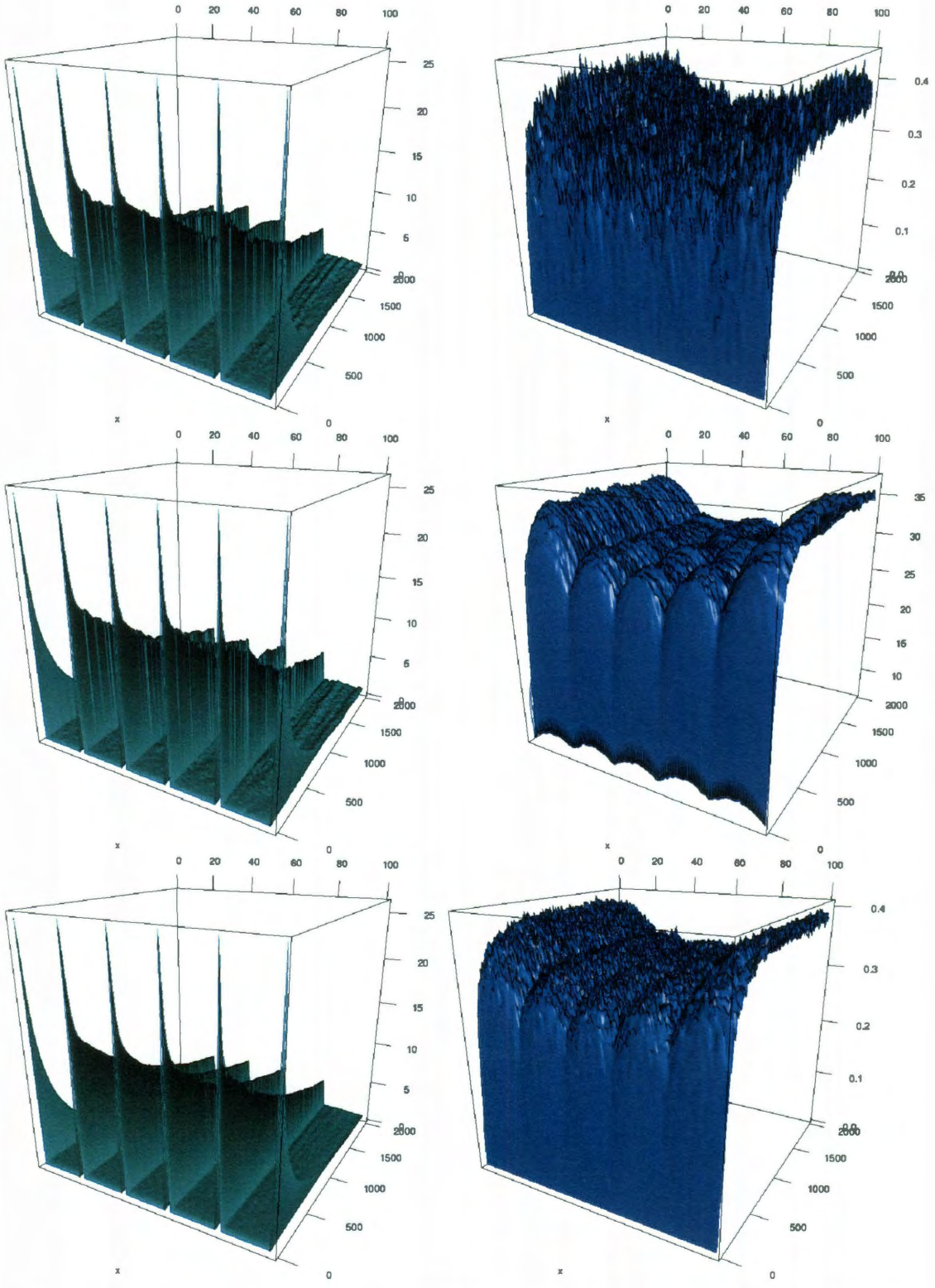


Figure D.4 : Stochastic reaction-diffusion system. Pre-cancerous cells. Mean and standard deviation of 10,000 runs. Initial conditions: end values of perturbed deterministic system rounded to nearest integer. Left: mean. Right: standard deviation.

Appendix E

Additional material for Chapter 9

E.1 Complete set of model reactions

$$\begin{aligned}
\mathrm{Vir} + \mathrm{UC} &\xrightarrow{\text{UC_infection_r / vir_avg}} \mathrm{IC}_1 \\
\mathrm{IC}_1 &\xrightarrow{\text{IC_deterioration_r}} \mathrm{IC}_2 \\
\mathrm{IC}_2 &\xrightarrow{\text{IC_deterioration_r}} \mathrm{IC}_3 \\
\mathrm{IC}_3 &\xrightarrow{\text{IC_ifn_production_r}} \mathrm{IC}_3 + \mathrm{Ifn} \\
\mathrm{IC}_3 &\xrightarrow{\text{IC_death_r / 4}} \mathrm{DC} \\
\mathrm{IC}_3 &\xrightarrow{\text{IC_deterioration_r}} \mathrm{IC}_4 \\
\mathrm{IC}_4 &\xrightarrow{\text{IC_ifn_production_r / 2}} \mathrm{IC}_4 + \mathrm{Ifn} \\
\mathrm{IC}_4 &\xrightarrow{\text{IC_vir_production_r / 2}} \mathrm{IC}_4 + \mathrm{Vir} \\
\mathrm{IC}_4 &\xrightarrow{\text{IC_death_r / 2}} \mathrm{DC} \\
\mathrm{IC}_4 &\xrightarrow{\text{IC_deterioration_r}} \mathrm{IC}_5 \\
\mathrm{IC}_5 &\xrightarrow{\text{IC_vir_production_r}} \mathrm{IC}_5 + \mathrm{Vir} \\
\mathrm{IC}_5 &\xrightarrow{\text{IC_death_r}} \mathrm{DC} \\
\mathrm{Ifn} + \mathrm{UC} &\xrightarrow{\text{UC_resistance_r / ifn_avg}} \mathrm{RC} \\
\mathrm{RC} &\xrightarrow{\text{RC_recover_r}} \mathrm{UC} \\
\mathrm{Ifn} &\xrightarrow{\text{ifn_degr_r}} \otimes \\
\mathrm{Vir} &\xrightarrow{\text{vir_degr_r}} \otimes
\end{aligned}$$

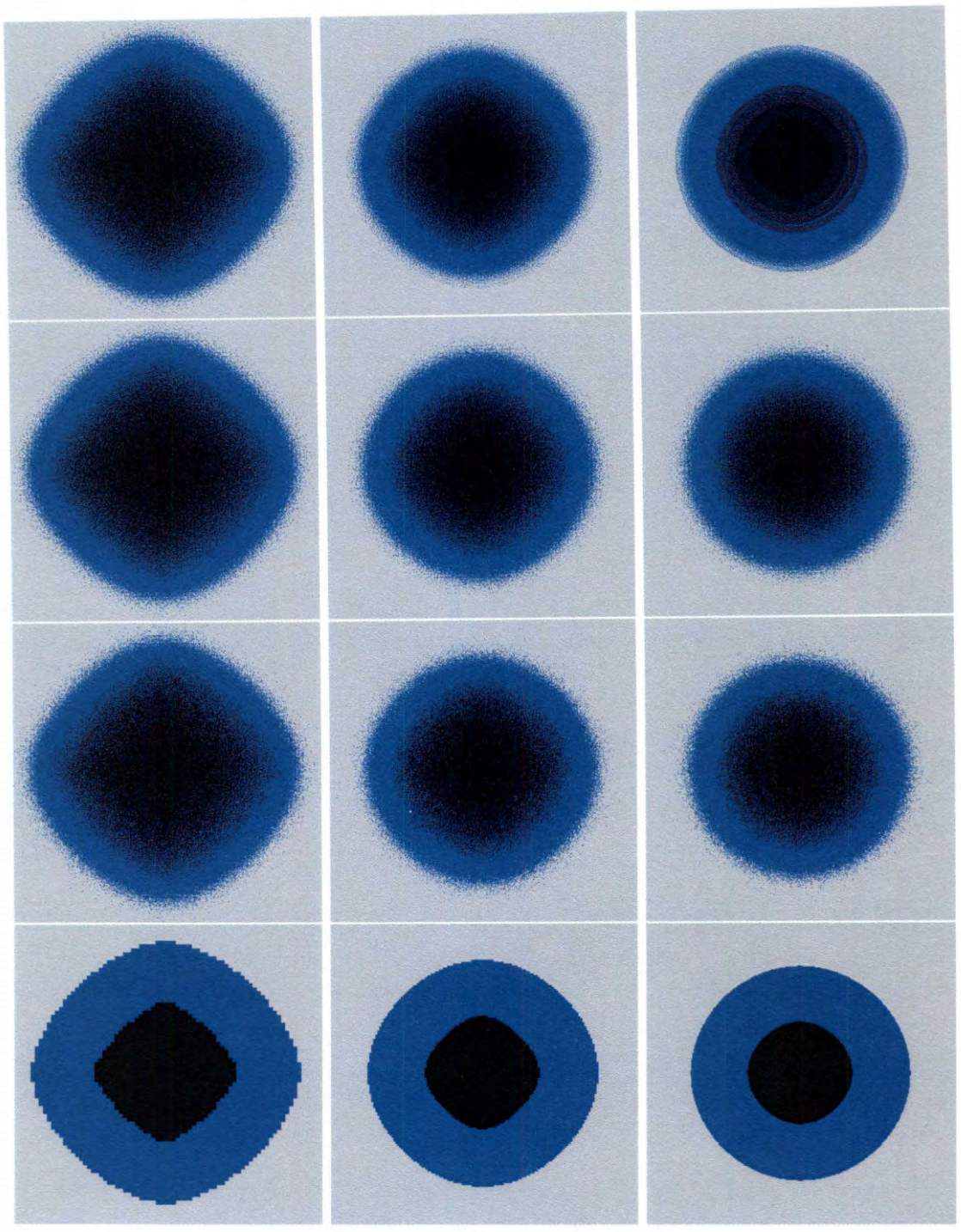


Figure E.1 : Different ways of presenting the same deterministic result, for different number of compartments. First column: grid of 140x140 compartments (64 cells per compartment). Second column: grid of 280x280 compartments (16 cells per compartemnt). Third column: grid of 560x560 compartments (4 cells per compartment). Each row is a different presentation of the same deterministic result (row 1: floor+maximum, row 2: floor+maximum aggregated, row 3: probability, row 4: maximum).

Appendix F

R code

F.1 Code for chapters 5

Listing F.1: helper.R

```
init <- function(place) {
  model <- list(transitions=0,
                place=place,
                places=length(place),
                h=list())
  assign("model", model, envir = .GlobalEnv)

  assign("L", array(0, dim <- c(1, model$places)), envir = .GlobalEnv)
  assign("R", array(0, dim <- c(1, model$places)), envir = .GlobalEnv)

  for (n in 1:model$places) {
    assign(place[n], n, envir = .GlobalEnv)
  }
}

atr <- function(trans.name=NULL) {
  model <- get("model",envir = .GlobalEnv)
  model$transitions <- model$transitions + 1
  if (!is.null(trans.name)) {
    assign(trans.name, model$transitions, envir = .GlobalEnv)
  }
  if (model$transitions == 1) {
    model$pre <- get("L", envir = .GlobalEnv)
    model$post <- get("R", envir = .GlobalEnv)
  } else {
    model$pre <- rbind(model$pre,get("L", envir = .GlobalEnv))
    model$post <- rbind(model$post,get("R", envir = .GlobalEnv))
  }
  model$h[[model$transitions]] <- get("h",envir = .GlobalEnv)
  assign("model", model, envir = .GlobalEnv)

  assign("L", array(0, dim <- c(1, model$places)), envir = .GlobalEnv)
  assign("R", array(0, dim <- c(1, model$places)), envir = .GlobalEnv)
}

load.cfn <- function(place, code) {
  for (i in 1:length(place)) {
    if (i == 1) {
      concat.places <- place[i]
    } else {
      concat.places <- paste(concat.places, ", ", place[i], sep="")
    }
  }
  this.tempfile <- basename(tempfile(pattern="tmp-cfn-", tmpdir="."))

```

```
  c.file <- paste("typedef enum {", concat.places, "} enum_places;\n",
      code,"\n",
                  sep = "")
  cat(c.file, file=paste(this.tempfile, ".c", sep= ""))
  cat(c.file)

  system.str <- paste("R CMD SHLIB ", this.tempfile, ".c", sep= "")
  if (.Platform$OS.type == "windows") {
    shell(system.str)
  } else {
    system(system.str)
  }
  dyn.load(paste(this.tempfile, .Platform$dynlib.ext, sep=""))
  getNativeSymbolInfo("cfn", PACKAGE=this.tempfile)$address
}

unload.cfns <- function()  {
  files <- list.files(path = ".",
                      pattern = "^tmp-cfn-[[:xdigit:]]*\\.(so|dll|c|o)$"
                      ,
                      full.names = F, recursive = F)
  for (file in files) {
    split.file <- strsplit(file, "\\.")[[1]]
    file.extension <- paste(".", split.file[2], sep="")
    if (file.extension == .Platform$dynlib.ext &
        is.loaded("cfn", split.file[1])) {
      cat(paste(file, "unloaded\n"))
      dyn.unload(file)
    }
    cat(paste(file, "deleted\n"))
    unlink(file)
  }
}
```

F.2 Examples of jumping patterns, for different boundary conditions

Listing F.2: Two-dimensional deterministic Neumann boundary conditions

```
jumps.per.place <- 2*(tot.rows*(tot.cols-1)+tot.cols*(tot.rows-1)) +
                   2*(tot.rows+tot.cols)
jumps <- array(0, dim=c(jumps.per.place, 3))
jumps[,3] <- 1
jump.idx <- 0
if (tot.cols > 1) {
  for (r in 1:tot.rows) {
    for (cl in 1:(tot.cols-1)) {
      jump.idx <- jump.idx + 1
      jumps[jump.idx, 1] <- (r-1)*tot.cols + cl
      jumps[jump.idx, 2] <- (r-1)*tot.cols + cl + 1
      if (cl == tot.cols-1) {
        jumps[jump.idx, 3] <- 2
        jump.idx <- jump.idx + 1
        jumps[jump.idx, 1] <- (r-1)*tot.cols + cl + 1
```

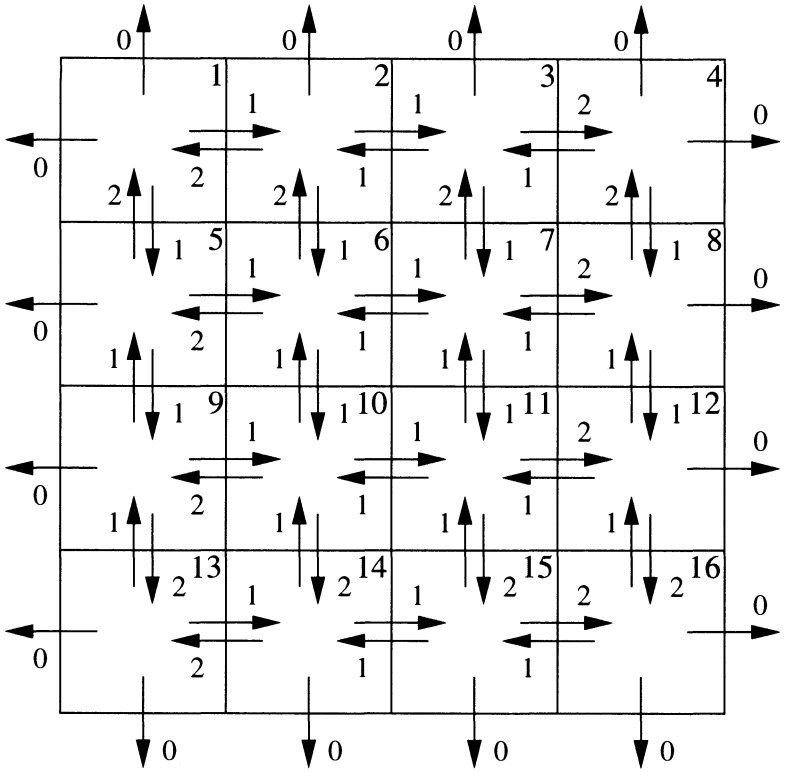


Figure F.1 : Jumping pattern for **deterministic** diffusive system with **Neumann** boundary conditions, two dimensional case.

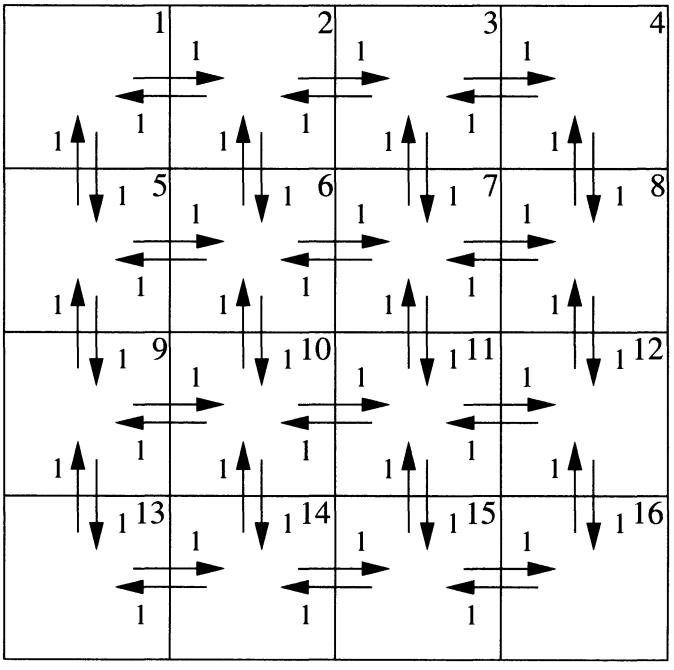


Figure F.2 : Jumping pattern for **stochastic** diffusive system with **Neumann** boundary conditions, two dimensional case.

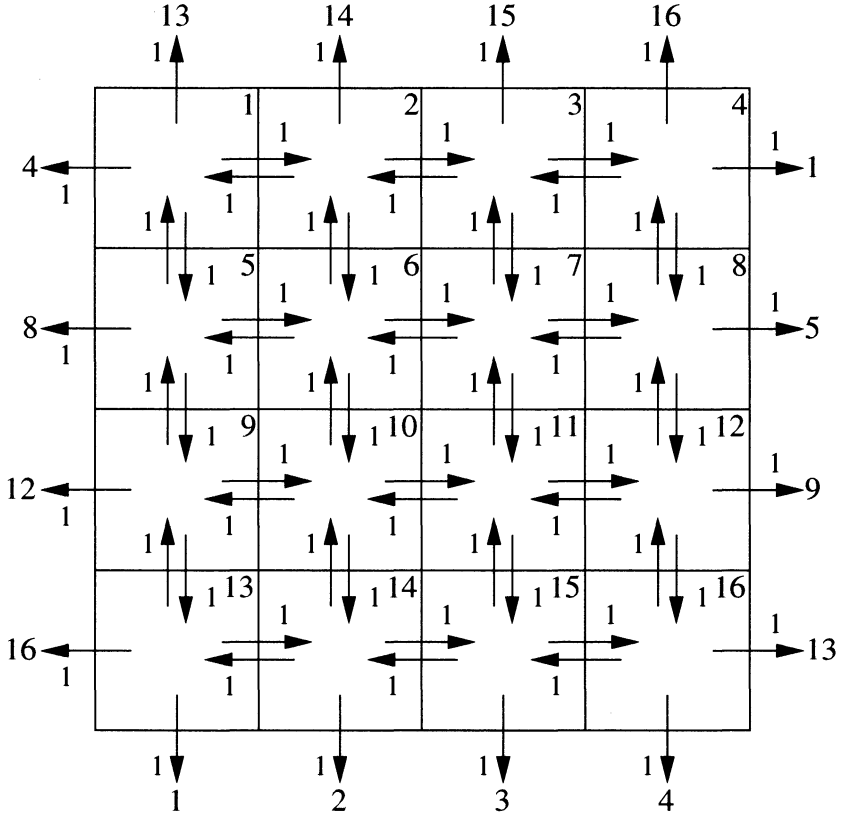


Figure F.3 : Jumping pattern for **deterministic** and **stochastic** diffusive system with **periodic** boundary conditions, two dimensional case.

```
16          jumps[jump.idx, 2] <- 0
17          jumps[jump.idx, 3] <- 0
18        }
19        jump.idx <- jump.idx + 1
20        jumps[jump.idx, 1] <- (r-1)*tot.cols + cl + 1
21        jumps[jump.idx, 2] <- (r-1)*tot.cols + cl
22        if (cl == 1) {
23          jumps[jump.idx, 3] <- 2
24          jump.idx <- jump.idx + 1
25          jumps[jump.idx, 1] <- (r-1)*tot.cols + cl
26          jumps[jump.idx, 2] <- 0
27          jumps[jump.idx, 3] <- 0
28        }
29      }
30    }
31  }
32  if (tot.rows > 1) {
33    for (cl in 1:tot.cols) {
34      for (r in 1:(tot.rows-1)) {
35        jump.idx <- jump.idx + 1
36        jumps[jump.idx, 1] <- (r-1)*tot.cols + cl
37        jumps[jump.idx, 2] <- (r)*tot.cols + cl
38        if (r == tot.rows-1) {
39          jumps[jump.idx, 3] <- 2
40          jump.idx <- jump.idx + 1
41          jumps[jump.idx, 1] <- (r)*tot.cols + cl
42          jumps[jump.idx, 2] <- 0
43          jumps[jump.idx, 3] <- 0
44        }
45        jump.idx <- jump.idx + 1
46        jumps[jump.idx, 1] <- (r)*tot.cols + cl
47        jumps[jump.idx, 2] <- (r-1)*tot.cols + cl
48        if (r == 1) {
49          jumps[jump.idx, 3] <- 2
50          jump.idx <- jump.idx + 1
51          jumps[jump.idx, 1] <- (r-1)*tot.cols + cl
52          jumps[jump.idx, 2] <- 0
53          jumps[jump.idx, 3] <- 0
54        }
55      }
56    }
57  }
58  JumpingPattern <- list(jumps)
```

Listing F.3: Two-dimensional stochastic Neumann boundary conditions

```
1  jumps.per.place <- 2*(tot.rows*(tot.cols-1)+tot.cols*(tot.rows-1))
2  jumps <- array(0, dim=c(jumps.per.place, 3))
3  jumps[,3] <- 1
4  jump.idx <- 0
5  for (r in 1:tot.rows) {
6    for (cl in 1:(tot.cols-1)) {
7      jump.idx <- jump.idx + 1
8      jumps[jump.idx, 1] <- (r-1)*tot.cols + cl
9      jumps[jump.idx, 2] <- (r-1)*tot.cols + cl + 1
10     jump.idx <- jump.idx + 1
11     jumps[jump.idx, 1] <- (r-1)*tot.cols + cl + 1
12     jumps[jump.idx, 2] <- (r-1)*tot.cols + cl
```

```
  }
}
for (cl in 1:tot.cols) {
  for (r in 1:(tot.rows-1)) {
    jump.idx <- jump.idx + 1
    jumps[jump.idx, 1] <- (r-1)*tot.cols + cl
    jumps[jump.idx, 2] <- r*tot.cols + cl
    jump.idx <- jump.idx + 1
    jumps[jump.idx, 1] <- r*tot.cols + cl
    jumps[jump.idx, 2] <- (r-1)*tot.cols + cl
  }
}
JumpingPattern <- list(jumps)
```

Listing F.4: Two-dimensional deterministic and stochastic periodic boundary conditions

```
jumps.per.place <- 4*tot.rows*tot.cols
jumps <- array(0, dim=c(jumps.per.place, 3))
jumps[,3] <- 1
jump.idx <- 0
for (r in 1:tot.rows) {
  for (cl in 1:(tot.cols-1)) {
    if (cl == 1) {
      jump.idx <- jump.idx + 1
      jumps[jump.idx, 1] <- (r-1)*tot.cols + 1
      jumps[jump.idx, 2] <- (r-1)*tot.cols + tot.cols
      jump.idx <- jump.idx + 1
      jumps[jump.idx, 1] <- (r-1)*tot.cols + tot.cols
      jumps[jump.idx, 2] <- (r-1)*tot.cols + 1
    }
    jump.idx <- jump.idx + 1
    jumps[jump.idx, 1] <- (r-1)*tot.cols + cl
    jumps[jump.idx, 2] <- (r-1)*tot.cols + cl + 1
    jump.idx <- jump.idx + 1
    jumps[jump.idx, 1] <- (r-1)*tot.cols + cl + 1
    jumps[jump.idx, 2] <- (r-1)*tot.cols + cl
  }
}
for (cl in 1:tot.cols) {
  for (r in 1:(tot.rows-1)) {
    if (r == 1) {
      jump.idx <- jump.idx + 1
      jumps[jump.idx, 1] <- cl
      jumps[jump.idx, 2] <- (tot.rows-1)*tot.cols + cl
      jump.idx <- jump.idx + 1
      jumps[jump.idx, 1] <- (tot.rows-1)*tot.cols + cl
      jumps[jump.idx, 2] <- cl
    }
    jump.idx <- jump.idx + 1
    jumps[jump.idx, 1] <- (r-1)*tot.cols + cl
    jumps[jump.idx, 2] <- r*tot.cols + cl
    jump.idx <- jump.idx + 1
    jumps[jump.idx, 1] <- r*tot.cols + cl
    jumps[jump.idx, 2] <- (r-1)*tot.cols + cl
  }
}
JumpingPattern <- list(jumps)
```

Bibliography

[1] Alberts, B., A. Johnson, J. Lewis, M. Raff, K. Roberts, and P. Walter: 2007, *Molecular Biology of the Cell.* Garland Science, 5 edition.

[2] Alexopoulou, L., A. C. Holt, R. Medzhitov, and R. A. Flavell: 2001, 'Recognition of double-stranded RNA and activation of NF-kappaB by Toll-like receptor 3.'. *Nature* **413**(6857), 732–8.

[3] Baril, M., M. Racine, F. Penin, and D. Lamarre: 2009, 'MAVS dimer is a crucial signaling component of innate immunity and the target of hepatitis C virus NS3/4A protease.'. *J Virol* **83**(3), 1299–311.

[4] Baum, A. and A. Garcia-Sastre: 2010, 'Induction of type I interferon by RNA viruses: cellular receptors and their substrates.'. *Amino Acids* **38**(5), 1283–99.

[5] Berg, H.: 1983, *Random Walks in Biology.* Princeton University Press.

[6] Bustin, S. A.: 2000, 'Absolute quantification of mRNA using real-time reverse transcription polymerase chain reaction assays.'. *J Mol Endocrinol* **25**(2), 169–93.

[7] Cao, Y., D. Gillespie, and L. Petzold: 2006, 'Efficient step size selection for the tau-leaping simulation method'. *The Journal of Chemical Physics* **124**(4), 044109.

[8] Cao, Y., D. Gillespie, and L. Petzold: 2007, 'Adaptive explicit-implicit tau-leaping method with automatic tau selection'. *The Journal of Chemical Physics* **126**(22), 224101.

[9] Chaouiya, C.: 2007, 'Petri net modelling of biological networks'. *Brief Bioinform* **8**(4), 210–219.

[10] Dai, W.: 2010, 'A new accurate finite difference scheme for Neumann (insulated) boundary condition of heat conduction'. *International Journal of Thermal Sciences* **49**(3), 571–579.

[11] Dormand, J. R. and P. J. Prince: 1980, 'A family of embedded Runge-Kutta formulae'. *Journal of Computational and Applied Mathematics* **6**(1), 19–26.

[12] Duca, K. A., V. Lam, I. Keren, E. E. Endler, G. J. Letchworth, I. S. Novella, and J. Yin: 2001, 'Quantifying viral propagation in vitro: toward a method for characterization of complex phenotypes.'. *Biotechnol Prog* **17**(6), 1156–65.

[13] Engblom, S., L. Ferm, A. Hellander, and P. Lötstedt: 2009, 'Simulation of Stochastic Reaction-Diffusion Processes on Unstructured Meshes'. *SIAM Journal on Scientific Computing* **31**(3), 1774–1797.

[14] Erban, R. and S. J. Chapman: 2009, 'Stochastic modelling of reaction-diffusion processes: algorithms for bimolecular reactions'. *Physical Biology* **6**(4), 046001 (18pp).

[15] Erban, R., S. J. Chapman, and P. Maini: 2007, 'A practical guide to stochastic simulations of reaction-diffusion processes'. *ArXiv e-prints*.

[16] Fitzgerald, K. A., S. M. McWhirter, K. L. Faia, D. C. Rowe, E. Latz, D. T. Golenbock, A. J. Coyle, S. Liao, and T. Maniatis: 2003, 'IKKepsilon and TBK1 are essential components of the IRF3 signaling pathway.'. *Nat Immunol* **4**(5), 491–6.

[17] Gack, M. U., Y. C. Shin, C. Joo, T. Urano, C. Liang, L. Sun, O. Takeuchi, S. Akira, Z. Chen, S. Inoue, and J. U. Jung: 2007, 'TRIM25 RING-finger E3 ubiquitin ligase is essential for RIG-I-mediated antiviral activity.'. *Nature* **446**(7138), 916–920.

[18] Gibson, M. and J. Bruck: 2000, 'Efficient exact stochastic simulation of chemical systems with many species and many channels'. *Journal of Physical Chemistry A* **104**(9), 1876–1889.

[19] Gillespie, D.: 1977, 'Exact stochastic simulation of coupled chemical reactions'. *J. Phys. Chem.* **81**(25), 2340–2361.

[20] Gillespie, D.: 2000, 'The chemical Langevin equation'. *The Journal of Chemical Physics* **113**(1), 297–306.

[21] Gillespie, D.: 2001, 'Approximate accelerated stochastic simulation of chemically reacting systems'. *The Journal of Chemical Physics* **115**(4), 1716–1733.

[22] Gillespie, D. and L. Petzold: 2003, 'Improved leap-size selection for accelerated stochastic simulation'. *The Journal of Chemical Physics* **119**(16), 8229–8234.

[23] Guo, B. and G. Cheng: 2007, 'Modulation of the interferon antiviral response by the TBK1/IKKi adaptor protein TANK.'. *J Biol Chem* **282**(16), 11817–26.

[24] Harris, L. and P. Clancy: 2006, 'A "partitioned leaping" approach for multiscale modeling of chemical reaction dynamics'. *J. Chem. Phys.* **125**, 144107.

[25] Haseltine, E. L.: 2005, 'Systems Analysis of Stochastic and Population Balance Models for Chemically Reacting Systems'. Ph.D. thesis, University of Wisconsin-Madison.

[26] Haseltine, E. L., V. Lam, J. Yin, and J. B. Rawlings: 2008, 'Image-guided modeling of virus growth and spread.'. *Bull Math Biol* **70**(6), 1730–48.

[27] Haseltine, E. L. and J. B. Rawlings: 2002, 'Approximate simulation of coupled fast and slow reactions for stochastic chemical kinetics'. *The Journal of Chemical Physics* **117**(15), 6959–6969.

[28] Haseltine, E. L., J. B. Rawlings, and J. Yin: 2005, 'Dynamics of viral infections: incorporating both the intracellular and extracellular levels'. *Computers & Chemical Engineering* **29**(3), 675–686. Computational Challenges in Biology.

[29] Higham, D. J.: 2001, 'An Algorithmic Introduction to Numerical Simulation of Stochastic Differential Equations'. *SIAM Review* **43**(3), 525–546.

[30] Hiscott, J.: 2007, 'Triggering the innate antiviral response through IRF-3 activation.'. *J Biol Chem* **282**(21), 15325–9.

[31] Hoffmann, A., A. Levchenko, M. L. Scott, and D. Baltimore: 2002, 'The IkappaB-NF-kappaB signaling module: temporal control and selective gene activation.'. *Science* **298**(5596), 1241–5.

[32] Hoops, S., S. Sahle, R. Gauges, C. Lee, J. Pahle, N. Simus, M. Singhal, L. Xu, P. Mendes, and U. Kummer: 2006, 'COPASI–a COmplex PAthway SImulator.'. *Bioinformatics* **22**(24), 3067–74.

[33] Isaacson, S. A. and C. S. Peskin: 2006, 'Incorporating Diffusion in Complex Geometries into Stochastic Chemical Kinetics Simulations.'. *SIAM J. Scientific Computing* **28**(1), 47–74.

[34] Kato, H., S. Sato, M. Yoneyama, M. Yamamoto, S. Uematsu, K. Matsui, T. Tsujimura, K. Takeda, T. Fujita, O. Takeuchi, and S. Akira: 2005, 'Cell type-specific involvement of RIG-I in antiviral response.'. *Immunity* **23**(1), 19–28.

[35] Kato, H., O. Takeuchi, S. Sato, M. Yoneyama, M. Yamamoto, K. Matsui, S. Uematsu, A. Jung, T. Kawai, K. J. Ishii, O. Yamaguchi, K. Otsu, T. Tsujimura, C. Koh, C. Reis e Sousa, Y. Matsuura, T. Fujita, and S. Akira: 2006, 'Differential roles of MDA5 and RIG-I helicases in the recognition of RNA viruses.'. *Nature* **441**(7089), 101–5.

[36] Kawai, T. and S. Akira: 2006, 'Innate immune recognition of viral infection.'. *Nat Immunol* **7**(2), 131–7.

[37] Kierzek, A. M.: 2002, 'STOCKS: STOChastic Kinetic Simulations of biochemical systems with Gillespie algorithm'. *Bioinformatics* **18**(3), 470–481.

[38] Kumar, H., T. Kawai, H. Kato, S. Sato, K. Takahashi, C. Coban, M. Yamamoto, S. Uematsu, K. J. Ishii, O. Takeuchi, and S. Akira: 2006, 'Essential role of IPS-1 in innate immune responses against RNA viruses.'. *J Exp Med* **203**(7), 1795–803.

[39] Li, H., Y. Cao, L. Petzold, and D. Gillespie: 2007, 'Algorithms and Software for Stochastic Simulation of Biochemical Reacting Systems.'. *Biotechnol Prog* **24**(1), 56–61.

[40] Liu, P., M. Jamaluddin, K. Li, R. P. Garofalo, A. Casola, and A. R. Brasier: 2007, 'Retinoic acid-inducible gene I mediates early antiviral response and Toll-like receptor 3 expression in respiratory syncytial virus-infected airway epithelial cells.'. *J Virol* **81**(3), 1401–11.

[41] Liu, P., M. Lu, B. Tian, K. Li, R. P. Garofalo, D. Prusak, T. G. Wood, and A. R. Brasier: 2009, 'Expression of an IKKgamma splice variant determines IRF3 and canonical NF-kappaB pathway utilization in ssRNA virus infection.'. *PLoS One* **4**(11), e8079.

[42] Livak, K. J. and T. D. Schmittgen: 2001, 'Analysis of relative gene expression data using real-time quantitative PCR and the 2(-Delta Delta C(T)) Method.'. *Methods* **25**(4), 402–8.

[43] Marciniak-Czochra, A. and M. Kimmel: 2006a, 'Dynamics of growth and signaling along linear and surface structures in very early tumors.'. *Computational & Mathematical Methods in Medicine* **7**(2/3), 189–213.

[44] Marciniak-Czochra, A. and M. Kimmel: 2006b, 'Reaction–diffusion approach to modeling of the spread of early tumors along linear or tubular structures'. *Journal of Theoretical Biology* **244**(3), 375–387.

[45] Marciniak-Czochra, A. and M. Kimmel: 2007, 'Modelling of early lung cancer progression: Influence of growth factor production and cooperation between partially transformed cells'. *Math. Mod. Meth. Appl. Sci.* **17S**, 1693–1719.

[46] Marciniak-Czochra, A. and M. Kimmel: 2008, 'Reaction-difusion model of early carcinogenesis: The effects of influx of mutated cells.'. *Mathematical Modelling of Natural Phenomena* **3**(7), 90–114.

[47] Marciniak-Czochra, A. and M. Ptashnyk: 2008, 'Derivation of a macroscopic receptor-based model using homogenization techniques'. *SIAM J. Math. Anal.* **40**(1), 215–237.

[48] Matsumoto, M. and T. Nishimura: 1998, 'Mersenne Twister: A 623-Dimensionally Equidistributed Uniform Pseudo-Random Number Generator'. *ACMTMCS: ACM Transactions on Modeling and Computer Simulation* **8**.

[49] Murata, T.: 1989, 'Petri nets: properties, analysis and applications'. *Proceedings of the IEEE* **77**(4), 541–580.

[50] Nichol, J. C. and H. F. Deutsch: 1948, 'Biophysical studies of blood plasma proteins; separation of gamma-globulin from the sera of various animals.'. *J Am Chem Soc* **70**(1), 80–3.

[51] Okubo, A., P. K. Maini, M. H. Williamson, and J. D. Murray: 1989, 'On the spatial spread of the grey squirrel in Britain.'. *Proc R Soc Lond B Biol Sci* **238**(1291), 113–25.

[52] Oshiumi, H., M. Matsumoto, S. Hatakeyama, and T. Seya: 2009, 'Riplet/RNF135, a RING finger protein, ubiquitinates RIG-I to promote

interferon-beta induction during the early phase of viral infection.'. *J Biol Chem* **284**(2), 807–17.

[53] Paszek, P.: 2007, 'Modeling stochasticity in gene regulation: characterization in the terms of the underlying distribution function.'. *Bull Math Biol* **69**(5), 1567–601.

[54] Paulsson, J., O. G. Berg, and M. Ehrenberg: 2000, 'Stochastic focusing: fluctuation-enhanced sensitivity of intracellular regulation.'. *Proc Natl Acad Sci U S A* **97**(13), 7148–53.

[55] Pearson, J. E.: 1993, 'Complex patterns in a simple system.'. *Science* **261**(5118), 189–92.

[56] Pineda-Krch, M.: 2008, 'GillespieSSA: Implementing the Gillespie Stochastic Simulation Algorithm in R'. *Journal of Statistical Software* **25**(12), 1–18.

[57] Porterfield, J. S., D. C. Burke, and A. C. Allison: 1960, 'An estimate of the molecular weight of interferon as measured by its rate of diffusion through agar.'. *Virology* **12**, 197–203.

[58] Puszynski, K., R. Bertolusso, and T. Lipniacki: 2009, 'Crosstalk between p53 and nuclear factor-κB systems: Pro- and anti-apoptotic functions of NF-κB'. *IET Systems Biology* **3**(5), 356–367.

[59] R Development Core Team: 2008, 'R: A Language and Environment for Statistical Computing'. R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0.

[60] Reisig, W.: 1985, *Petri nets: an introduction.* New York, NY, USA: Springer-Verlag New York, Inc.

[61] Schwartz, R.: 2008, *Biological Modeling and Simulation: A Survey of Practical Models, Algorithms, and Numerical Methods.* The MIT Press.

[62] Sen, G. C. and S. N. Sarkar: 2005, 'Transcriptional signaling by double-stranded RNA: role of TLR3.'. *Cytokine Growth Factor Rev* **16**(1), 1–14.

[63] Sharma, S., B. R. tenOever, N. Grandvaux, G. Zhou, R. Lin, and J. Hiscott: 2003, 'Triggering the interferon antiviral response through an IKK-related pathway.'. *Science* **300**(5622), 1148–51.

[64] Slepoy, A., A. P. Thompson, and S. J. Plimpton: 2008, 'A constant-time kinetic Monte Carlo algorithm for simulation of large biochemical reaction networks.'. *J Chem Phys* **128**(20), 205101.

[65] Sun, S. C., P. A. Ganchi, D. W. Ballard, and W. C. Greene: 1993, 'NF-kappa B controls expression of inhibitor I kappa B alpha: evidence for an inducible autoregulatory pathway.'. *Science* **259**(5103), 1912–5.

[66] tenOever, B. R., S. Sharma, W. Zou, Q. Sun, N. Grandvaux, I. Julkunen, H. Hemmi, M. Yamamoto, S. Akira, W. Yeh, R. Lin, and J. Hiscott: 2004, 'Activation of TBK1 and IKKvarepsilon kinases by vesicular stomatitis virus infection and the role of viral ribonucleoprotein in the development of interferon antiviral immunity.'. *J Virol* **78**(19), 10636–49.

[67] Wagner, M., M. Schmid, S. Juretschko, K. H. Trebesius, A. Bubert, W. Goebel, and K. H. Schleifer: 1998, 'In situ detection of a virulence factor mRNA and 16S rRNA in Listeria monocytogenes.'. *FEMS Microbiol Lett* **160**(1), 159–68.

[68] Ware, B. R., T. Raj, W. H. Flygare, J. A. Lesnaw, and M. E. Reichmann: 1973, 'Molecular weights of vesicular stomatitis virus and its defective particles by laser light-scattering spectroscopy.'. *J Virol* **11**(1), 141–5.

[69] Wilkinson, D.: 2006, *Stochastic modelling for systems biology*, Mathematical and Computational Biology Series. Chapman & Hall/CRC.

[70] Yamamoto, M., S. Sato, K. Mori, K. Hoshino, O. Takeuchi, K. Takeda, and S. Akira: 2002, 'Cutting edge: a novel Toll/IL-1 receptor domain-containing adapter that preferentially activates the IFN-beta promoter in the Toll-like receptor signaling.'. *J Immunol* **169**(12), 6668–72.

[71] Zhao, T., L. Yang, Q. Sun, M. Arguello, D. W. Ballard, J. Hiscott, and R. Lin: 2007, 'The NEMO adaptor bridges the nuclear factor-kappaB and interferon regulatory factor signaling pathways.'. *Nat Immunol* **8**(6), 592–600.